МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра программной инженерии

Специальность 1-40 05 01 03 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТ:**

по дисциплине «Объектно-ориентированные технологии программирования»

Тема «Программное средство «Тесты ПДД»

Исполнитель

студент 3 курса группы 2 Скалкович Станислав Леонидович

(Ф.И.О.)

Руководитель работы преп.-стажер Якунович А.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Смелов В.В.

(подпись)

Минск 2023

**Содержание**

[Введение 4](#_Toc154159384)

[1 Аналитический обзор литературы и формирование требований 5](#_Toc154159385)

[1.1 Анализ прототипов 5](#_Toc154159386)

[1.2 Требования к проекту 7](#_Toc154159387)

[1.3 Вывод 7](#_Toc154159388)

[2 Анализ и проектирование архитектуры приложения 8](#_Toc154159389)

[2.1 Описание средств разработки 8](#_Toc154159390)

[2.1.1 Rider 2023 и Microsoft Visual Studio 2022 8](#_Toc154159391)

[2.1.2 Программная платформа .NET 7 8](#_Toc154159392)

[2.1.3 Язык программирования C# 8](#_Toc154159393)

[2.1.4 Технология WPF 9](#_Toc154159394)

[2.1.5 Расширяемый язык разметки XAML 9](#_Toc154159395)

[2.1.6 Технология Entity Framework 9](#_Toc154159396)

[2.1.7 SQLite 9](#_Toc154159397)

[2.1.8 Паттерн MVVM 10](#_Toc154159398)

[2.2 Спецификация функциональных требований к программному средству 10](#_Toc154159399)

[2.3 Спецификация функциональных требований 10](#_Toc154159400)

[2.4 Вывод 11](#_Toc154159401)

[3 Проектирование программного средств 12](#_Toc154159402)

[3.1 Общая структура 12](#_Toc154159403)

[3.1.1 Структура клиентской части 12](#_Toc154159404)

[3.1.2 Структура серверной части 13](#_Toc154159405)

[3.2 Взаимоотношение между классами 13](#_Toc154159406)

[3.3 Модель базы данных 13](#_Toc154159407)

[3.4 Вывод 14](#_Toc154159408)

[4 Разработка функциональной модели и модели данных 15](#_Toc154159409)

[4.1 Основные модели программного средства 15](#_Toc154159410)

[4.2 Активация продукта 15](#_Toc154159411)

[4.3 Выбор случайного теста 15](#_Toc154159412)

[4.4 Выбор случайного теста по теме 16](#_Toc154159413)

[4.5 Прохождение теста 16](#_Toc154159414)

[4.6 Вывод результатов теста 16](#_Toc154159415)

[4.7 Возможность просмотреть описание к вопросу 16](#_Toc154159416)

[4.8 Вывод 16](#_Toc154159417)

[5 Тестирование, проверка работоспособности и анализ полученных результатов 17](#_Toc154159418)

[5.1 Тестирование активации ключа 17](#_Toc154159419)

[5.2 Вывод 18](#_Toc154159420)

[6 Руководство по установке и использованию 18](#_Toc154159421)

[Заключение 23](#_Toc154159422)

[Список использованных источников 24](#_Toc154159423)

[Приложение А 25](#_Toc154159424)

[Приложение Б 26](#_Toc154159425)

[Приложение В 27](#_Toc154159426)

[Приложение Г 28](#_Toc154159427)

# Введение

Данный курсовой проект посвящен разработке программного средства, предоставляющего тесты ПДД. Программное средство представляет собой графический интерфейс для работы с тестами. К основным функциональным возможностям программного средства можно отнести:

Предоставление тестов ПДД: пользователь может проходить тесты для совершенствования и контроля своих знаний.

Просмотр результатов теста: пользователь может просмотреть таблицу результатов, после завершения теста, для ознакомления со своими результатами.

Работа над ошибками: пользователь сможет вернуться к вопросу, вызвать подсказку, и ознакомиться с разъяснением к вопросу.

Технические детали проекта включают в себя:

Сервер на ASP.NET Core Web API: для обеспечения серверной части для активации программного средства. Сервер реализован на технологии ASP.NET Core Web API. Это обеспечит кроссплатформенность, безопасность и производительность сервера.

База данных Sqlite: для хранения информации о вопросах ПДД и ключах активации, будет использоваться легковесная база данных SQLite, которая обеспечит удобное обновление данных о тестах и не требует дополнительного севера.

Клиентская часть на WPF: для создания клиентской части программного средства будет использован Windows Presentation Foundation (WPF). WPF предоставляет мощные средства для разработки интерфейсов и визуализации данных.

Подключение к серверу: клиент программного средства будет взаимодействовать с сервером через API, по протоколу HTTP, обеспечивая проверку ключа активации и возвращение ответа о состоянии проверки ключа.

Проект будет спроектирован с учетом принципов ООП, SOLID.

# 1 Аналитический обзор литературы и формирование требований

## Анализ прототипов

Были проанализированы цели и задачи, поставленные в данном курсовом проекте, а также рассмотрены аналогичные примеры их решений. На основании анализа всех достоинств и недостатков данных альтернативных решений были сформулированы требования к данному программному средству.

Первый аналог – ADrive рисунок 1.1.

ADrive - это онлайн-платформа, предлагающая тренажеры для подготовки к экзаменам на права в различных странах, включая Россию. Вот некоторые особенности и преимущества ADrive:

Широкий выбор языков: ADrive предоставляет возможность изучения ПДД на разных языках, что делает его удобным для пользователей со всего мира.

- Интерактивные уроки: платформа предлагает интерактивные учебные материалы, включая вопросы и задания, чтобы помочь вам понять и запомнить правила дорожного движения.

- Практические тесты: ADrive предлагает различные практические тесты, которые помогут вам оценить свои знания и подготовиться к экзамену на получение водительского удостоверения.

- Обратная связь: после прохождения тестов платформа предоставляет обратную связь и объяснения по неправильным ответам, что поможет вам исправить ошибки и улучшить свои знания.

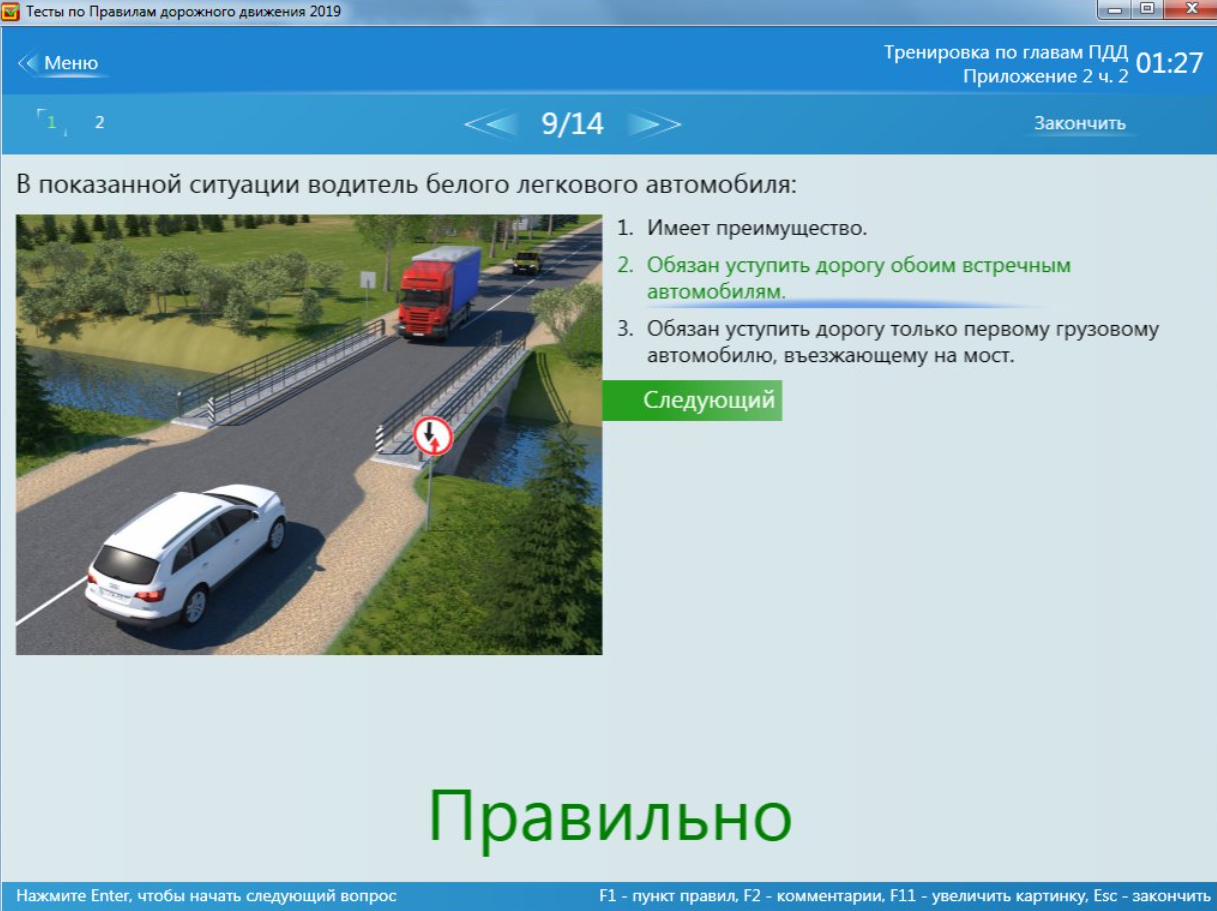


Рисунок 1.1 – Приложение ADrive

Второй аналог – ДОСААФ рисунок 1.2.

ДОСААФ (Добровольное Общество Содействия Армии, Авиации и Флоту) - это российская общественная организация, которая также предлагает тренажеры ПДД для подготовки к экзаменам на водительское удостоверение. Вот некоторые особенности и преимущества ДОСААФ:

Курсы обучения: ДОСААФ предлагает оффлайн-курсы обучения, включающие теоретические занятия и практическую подготовку на специальных тренажерах.

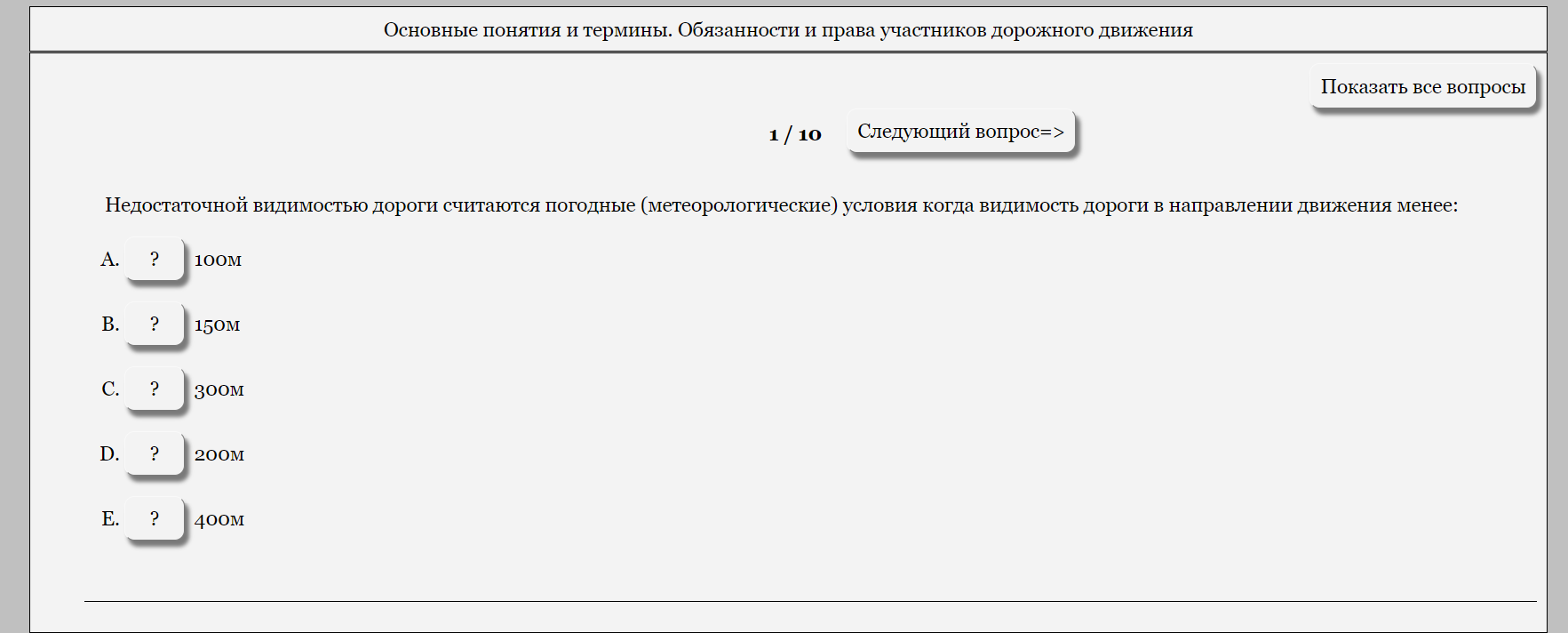
- Опытные инструкторы: организация предлагает обучение под руководством опытных инструкторов, которые помогут вам освоить правила дорожного движения и приобрести навыки вождения.

- Подготовка к экзаменам: ДОСААФ организует практическую подготовку к экзаменам на получение водительского удостоверения и предоставляет возможность сдачи экзаменов на их территории.

- Дополнительные услуги: организация также предоставляет дополнительные услуги, такие как медицинское освидетельствование и консультации специалистов.

При разработке своего приложения тренажера ПДД, можно учесть следующие аспекты, реализованные в аналогах:

* Практические тесты: Разработайте практические тесты, которые позволят пользователям проверить свои знания ПДД и оценить свою готовность к экзамену на получение водительского удостоверения. Предоставляйте обратную связь и объяснения к неправильным ответам для повышения эффективности обучения.
* Оффлайн-обучение: Предоставьте возможность пользователям обучаться в оффлайн-режиме, чтобы они могли изучать ПДД в любое удобное для них время и в любом месте.
* Удобство использования: Обратите внимание на дизайн и пользовательский интерфейс приложения, чтобы оно было интуитивно понятным и удобным в использовании. Разделите материалы на логические разделы, предоставьте навигацию и поиск, чтобы пользователи могли легко найти нужную информацию.



* Рисунок 1.2 – Приложение dosaaf

## 1.2 Требования к проекту

Обзор аналогов тестов ПДД позволяет сформулировать ключевые требования к разрабатываемому программному средству. Программное средство должно поддерживать следующие функции:

1. Активация программного средства по ключу:

* Активация по ключу необходима для того, чтобы программным средством могли пользоваться только те, кто купил ключ для активации, в противном случае приложение не будет запускаться и предоставлять доступ к своему функционалу. Ключ отправляется с формы на сервер, который возвращает хэш ключа, приложение проверяет наличие и корректность хэша, и при положительном исходе запускает основную часть, в противном, выдает ошибку о некорректности ключа.

1. Дешифрование данных:

* В базе данных вопросы зашифрованы алгоритмом RSA для защиты от копирования данных. В приложении встроен приватный ключ, с помощью которого приложение расшифровывает тест вопросов.

1. Прохождение теста:

* Пользователь выбирает контроль по случайному, после этого запускается тестирование по случайным вопросам в базу данных.
* Пользователь выбирает контроль по теме, после этого запускается тестирование по случайным вопросам определенной темы из базы данных.

1. Просмотр результатов теста:

* После завершения теста, пользователю открывается краткий результат теста, где представлен результат теста, и количество правильных и неправильных ответов. Далее открывается таблица с подробными результатами и ответами пользователя, с данной таблицы можно перейти на определенный вопрос для работы над своей ошибкой.

1. Работа над ошибками:

* В данной части программы представлен вопрос, указанный правильный ответ и ответ пользователя, результат ответа, а также при нажатии на пробел, показывается подробный ответ на вопрос основывающейся на ПДД.

Эти требования определяют основной функционал программного средства, которое будет разработано в рамках данного проекта. Реализация этих функций позволит пользователям успешно проходить тесты, работать над своими ошибками и закреплять знания по ПДД.

## 1.3 Вывод

Итого, был проведен аналитический обзор программный среств предоставляющих тесты ПДД, которые уже существуют на рынке. Этот обзор позволил определить основные характеристики и функциональные возможности, которые необходимо предусмотреть в разрабатываемой системе. Также были определены функциональные требования приложения.

# 2 Анализ и проектирование архитектуры приложения

## 2.1 Описание средств разработки

При разработке приложения были использованы:

* интегрированные среды разработки Rider 2023 и Microsoft Visual Studio 2022;
* программная платформа .NET 7.0;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* технология Entity Framework;
* SQLite;
* ASP .NET CORE.

### 2.1.1 Rider 2023 и Microsoft Visual Studio 2022

Rider — кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Rider поддерживает .NET Framework, новую платформу .NET Core и проекты на основе Mono. IDE позволяет разрабатывать десктопные приложения, .NET-сервисы и библиотеки, игры на движке Unity, мобильные приложения Xamarin, веб-приложения ASP.NET и ASP.NET Core.

Microsoft Visual Studio 2022 – это интегрированная среда разработки для написания, отладки и сборки кода, а также последующей публикации приложений. Данный продукт позволяет разрабатывать не только консольные, но и десктопные приложения, с использованием таких технологий, как WinForms или WPF.

### 2.1.2 Программная платформа .NET 7

Платформа .NET - это программный фреймворк, разработанный компанией Microsoft, который в основном работает на операционной системе Microsoft Windows. Он предоставляет среду выполнения для запуска приложений и набор библиотек, которые используются для создания программных приложений. Фреймворк .NET состоит из нескольких компонентов, включая общую языковую среду (Common Language Runtime - CLR), библиотеку классов .NET Framework и различные инструменты разработки.

### 2.1.3 Язык программирования C#

В качестве языка программирования используется C# – основной язык разработки в .NET. Язык объектно-ориентированный, имеет строгую статическую типизацию, поддерживает перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения. Используется как основной язык в технологии WPF.

### 2.1.4 Технология WPF

Для предоставления пользовательского интерфейса и разграничения дизайна и бизнес-логики используется технология Microsoft WPF – аналог WinForms, система для построения клиентских приложений Windows с возможностями взаимодействия с пользователем и графическая подсистема в составе .NET, использующая язык разметки XAML.

### 2.1.5 Расширяемый язык разметки XAML

WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление. XAML представляет собой язык декларативного описания интерфейса, основанный на XML.

### 2.1.6 Технология Entity Framework

Entity Framework (EF) - это современный ORM (Object-Relational Mapping) фреймворк, разработанный для .NET Framework. Он предоставляет более высокоуровневый способ работы с базами данных в сравнении с технологией ADO.NET.

Entity Framework (EF) предоставляет следующие возможности: абстракция данных, ленивая загрузка, поддержка различных источников данных, интеграция с LINQ, генерация схемы, отслеживание изменений, миграции, простота использования, уменьшение необходимости вручную писать SQL-запросы, поддержка асинхронных операций, управление транзакциями, облегчение тестирования.

### 2.1.7 SQLite

SQLite — это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы.

Понятие «встраиваемый» означает, что СУБД не использует парадигму клиент-сервер. Движок SQLite — не отдельно работающий процесс, с которым взаимодействует программа, а библиотека. Программа компонуется с ней, и движок служит составной частью программы. В качестве протокола обмена применяются вызовы функций (API) библиотеки SQLite.

### 2.1.8 Паттерн MVVM

MVVM (Model-View-ViewModel) - это паттерн архитектуры, используемый в разработке программного обеспечения для разделения пользовательского интерфейса (View) от бизнес-логики (Model) и связывания их через промежуточный слой ViewModel.

Представление (View) отвечает за отображение данных и взаимодействие с пользователем. Оно не содержит бизнес-логику и должно быть максимально независимым от модели.

ViewModel представляет промежуточный слой между View и Model. Он содержит логику, необходимую для обработки пользовательских действий, обновления данных в модели и уведомления View об изменениях. ViewModel предоставляет свойства и команды, которые привязываются к элементам пользовательского интерфейса во View, позволяя им отображать данные и реагировать на действия пользователя.

MVVM позволяет достичь разделения ответственности между компонентами приложения, облегчает тестирование и повышает переиспользуемость кода. Он также способствует лучшей поддержке параллельной разработки пользовательского интерфейса и бизнес-логики, так как разработчики могут работать независимо над своими частями приложения.

## 2.2 Спецификация функциональных требований к программному средству

Программное средство “Тесты ПДД” должно предоставлять следующие функциональные возможности:

Для пользователя:

* Активация продукта;
* Выбор случайного теста;
* Выбор случайного теста по теме;
* Прохождение теста;
* Вывод результатов теста;
* Возможность просмотреть описание к вопросу (работа над ошибками)

## 2.3 Спецификация функциональных требований

Для функциональности ПС необходимо создание базы данных для хранения информации приложения. Подробно база данных описано в следующем разделе.

В программном средстве необходимо реализовать выборку тестов, для этого необходимо определить базу данных и определить use-case диаграмму для определения данных в базе данных для функционала пользователя.

Описание функциональности программного средства представлено на диаграмме вариантов использования, изображенной на рисунке 2.1.

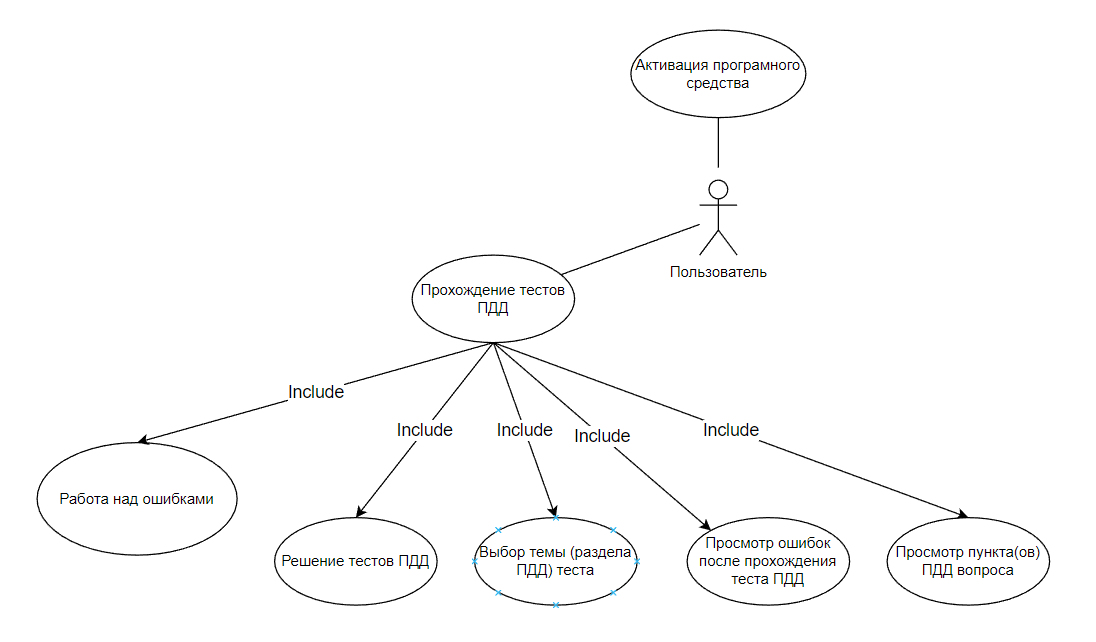


Рисунок 2.1 – Диаграмма вариантов использования

Описание структуры данных программного средства представлено на схеме базы данных, изображенной на рисунке 2.2.

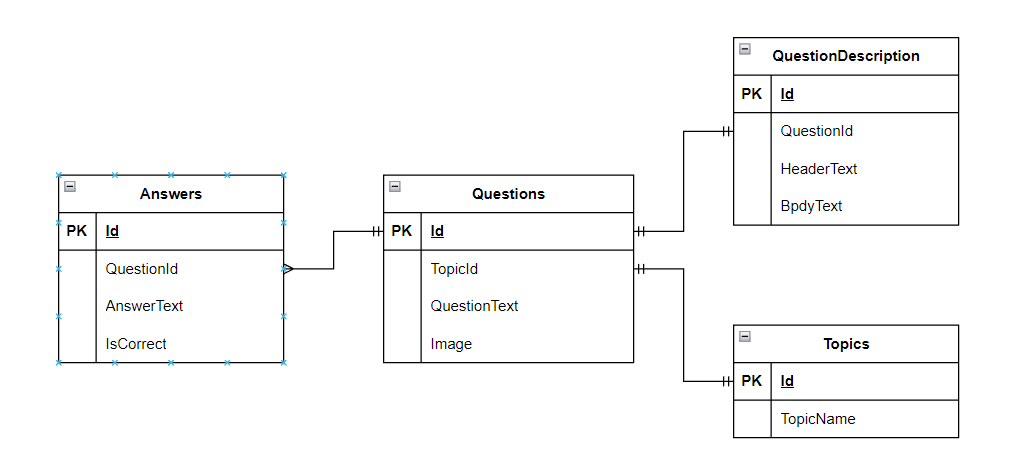


Рисунок 2.2 – Схема БД

На основе составленной use-case диаграммы мы можем четко определить функционал необходимый для реализации программного средства, а также, определить необходимые данные для хранения и реализовать схему базы данных.

## 2.4 Вывод

В данном разделе были выбраны основные средства разработки для программного средства, включающие технологии как для клиентской, так и для серверной частей. Особое внимание уделено определению структуры и инструментов, обеспечивающих эффективное разделение функциональности между клиентом и сервером.

# 3 Проектирование программного средств

## 3.1 Общая структура

Общая структура программного средства состоит из двух основных частей: клиентской и серверной. Ниже приведено подробное описание структуры этих частей.

### 3.1.1 Структура клиентской части

Программное средство Тесты ПДД для клиентской части имеет следующею структуру, представленную на рисунке 3.1.

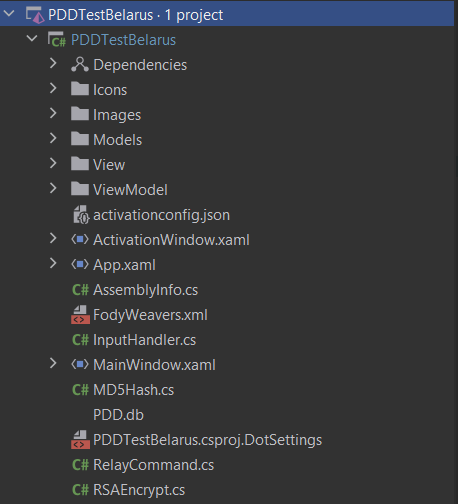


Рисунок 3.1 – Структура клиентской части проекта

Описание структуры основных папок и файлов клиентской части проекта представлено в таблице 3.1.

Таблица 3.1 – Описание клиенсткой структуры папок и файлов проекта

|  |  |
| --- | --- |
| Имя пакета | Содержание |
| Папка Models | Здесь описаны модели, с которыми происходит вся работа в приложении. |
| Папка View | В этой папке храняться файлы с xaml-разметкой, то есть окна и страницы. |
| Папка ViewModel | В данной папке хранятся все классы для взаимодействия графической составляющей и бизнес логики приложения. |
| Папка Icons | Иконки использующиеся в приложении. |
| Папка Images | В данной папке хранятся все изображения для тестов. |

В целом, описание структуры проекта позволяет лучше понимать, как устроено программное средство и какие компоненты в нем присутствуют.

### 3.1.2 Структура серверной части

Серверная часть программного средства имеет следующею структуру, представленную на рисунке 3.5.

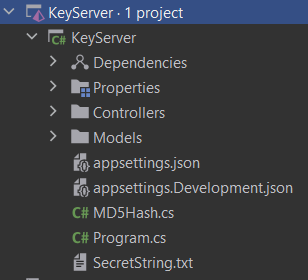


Рисунок 3.5 – Структура серверной части проекта

Описание структуры основных папок и файлов серверной части проекта представлено в таблице 3.5.

Таблица 3.5 – Описание клиенсткой структуры папок и файлов проекта

|  |  |
| --- | --- |
| Имя | Содержание |
| Папка Controllers | В данной папке содержатся классы, содержащие методы (действия), которые обрабатывают HTTP-запросы |
| Папка Models | Здесь описаны модели, с которыми происходит вся работа в приложении. |

## 3.2 Взаимоотношение между классами

Для визуализации взаимосвязей между классами используется диаграмма UML – графическое представление набора элементов, изображаемое чаще всего в виде связанного графа с вершинами (сущностями) и ребрами (отношениями).

Для представления внутренней структуры программы в виде классов и связей между ними используется диаграмма классов. Диаграмма классов представлена в приложении А.

## 3.3 Модель базы данных

Для успешной реализации целей по созданию и функционированию программного средства, была разработана база данных, использующая систему управления реляционными базами данных SQLite. Структура базы данных включает в себя пять основных таблиц: Quetions, QuestionDescription, Topics, Answers. База данных обеспечивает хранение и управление информацией о вопросах, подсказкам к вопросу, ответы и темы вопросов. Подробности о структуре базы данных и скрипт ее создания представлены в приложении Г.

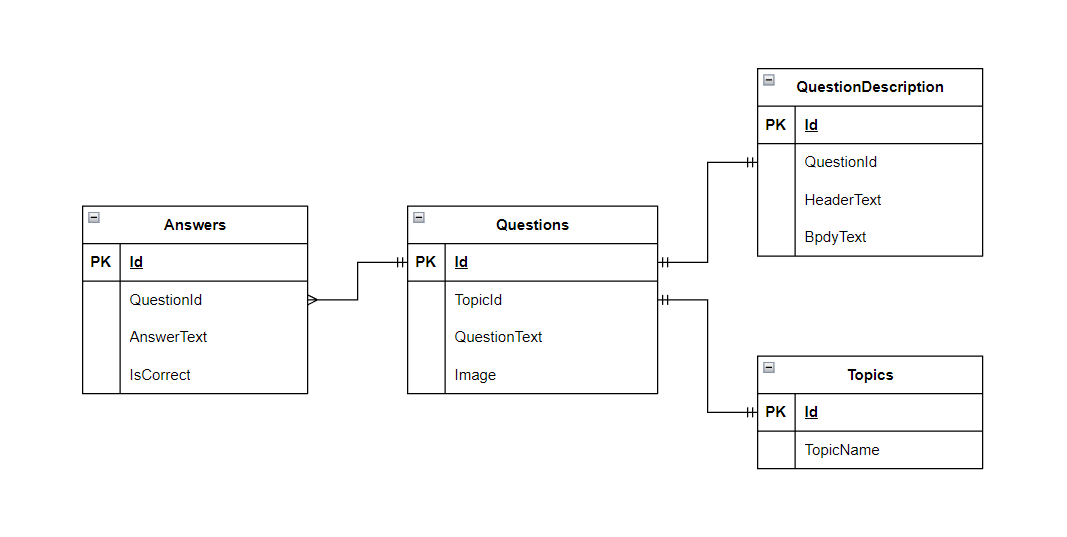


Рисунок 3.8 – База данных PDD

## 3.4 Вывод

В результате проектирования программного средства была разработана общая структура, включая подробное описание клиентской и серверной частей, а также взаимоотношение между классами. Определены основные принципы взаимодействия между компонентами приложения, что обеспечивает эффективное функционирование системы. Кроме того, была представлена модель базы данных, оптимизированная для хранения и управления вопросов к тестам и связанными с ними данными. Этот раздел играет ключевую роль в обеспечении стабильной и эффективной работы.

# 4 Разработка функциональной модели и модели данных

## 4.1 Основные модели программного средства

Для выполнения технических задач программного средства должны быть реализованы следующие функции и соответствующие им классы и методы:

* Активация продукта;
* Выбор случайного теста;
* Выбор случайного теста по теме;
* Прохождение теста;
* Вывод результатов теста;
* Возможность просмотреть описание к вопросу (работа над ошибками)

## 4.2 Активация продукта

Для графической части активации создано окно ActivationWindow.xaml, и вся логика работы определена в классе ActivationWindow.xaml.cs, а также в MainViewModel.cs

При первом запуске приложения в файл authorizeconfig.json записываеться уникальный uuid, который в последсвии будеит использоваться для получения хэша доступа.

После ввода и отправки ключа на сервер сервер проверяет, есть ли такой ключ и если да то возвращает уникальный хэш доступа, иначе высылает отрицательный ответ.

После ввода верного ключа в файл authorizeconfig.json записывается уникальный хэш отправленный сервером. Благодаря чему приложение при следующих запусках будет понимать что оно активировано и открывать окно MainView.xaml.

## 4.3 Выбор случайного теста

Для графической части активации создано окноHomePage.xaml, и вся логика работы определена в классе HomePage.xaml.cs, а также в MainViewModel.cs.

В данном окне при выборе 1-ого пункта меню выполняеться запрос в БД для получения 10 случайных вопросов, и запускается страница TestPage.xaml куда передается список с данными вопросов. Чтобы выйти из данного окна необходимо нажать клавишу ESC.

## 4.4 Выбор случайного теста по теме

Для графической части активации создано окноHomePage.xaml, и вся логика работы определена в классе HomePage.xaml.cs, а также в MainViewModel.cs. Чтобы выйти из данного окна необходимо нажать клавишу ESC.

В данном окне при выборе 2-ого пункта меню открывается окно TopicsPage.xaml, где пользователь выбирает определенную тему в последствии выполняется запрос в БД для получения 10 случайных вопросов по выбранной теме, и запускается страница TestPage.xaml куда передается список с данными вопросов. Чтобы выйти из данного окна необходимо нажать клавишу ESC.

## 4.5 Прохождение теста

После запуска страницы TestPage.xaml запускается элемент Question.xaml в который передается 1-ый вопрос, с помощью цифр клавиатуры пользователь может выбрать ответ, очистить поле ввода можно используя клавишу backspace. После подтверждение ответа формируется следующий обьект Question.xaml, если вопросы закончились, то формируется и запускается страница Results.xaml. Чтобы выйти из данного окна необходимо нажать клавишу ESC.

## 4.6 Вывод результатов теста

На основе пользовательских ответов и данных о вопросах формируется элемент ResultTable.xaml, где выводиться список пользовательских ответов и их правильность, используя стрелочки можно выбрать интересующий вопрос и перейти нему для просмотра своей ошибки. Чтобы выйти из данного окна необходимо нажать клавишу ESC.

## Возможность просмотреть описание к вопросу

На основе выбранного вопроса создается страница AfterTestPage.xaml, где используется компонент Question.xaml, нажав пробел появится окно с пояснениями к вопросу, изучив которые, можно понять свою ошибку, и запомнить алгоритм мыслей для следующей попытки. Чтобы выйти из данного окна необходимо нажать клавишу ESC.

## Вывод

В ходе реализации программного средства были активно разработаны и интегрированы основные классы, обеспечивающие корректное выполнение ключевых функций. Реализованы функционалы активация продукта, Выбор случайного теста, выбор теста по теме, прохождение теста, просмотр результатов теста, просмотр описания к вопросу. Все функции прошли тщательное тестирование, подтверждающее их надежную работу, что обеспечивает пользовательский опыт без сбоев и согласно ожиданиям пользователей.

# 5 Тестирование, проверка работоспособности и анализ полученных результатов

## Тестирование активации ключа

Для начала тестирования была проведена попытка ввода данных в поле активации ключа таких как цифры, символы, строки. При вводе данных не по шаблону показывает ошибку под текстовым полем (Рисунок 5.6), таким образом невозможно ввести ключ не по шаблону.

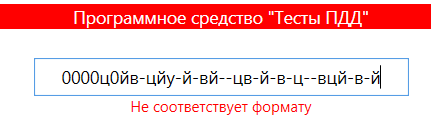


Рисунок 5.6 – Результат c пустыми полями

Далее проводилась проверка системы активации продукта, при отсутствии интернета или недоступности сервера выводиться соответствующее сообщение (Рисунок 5.7)

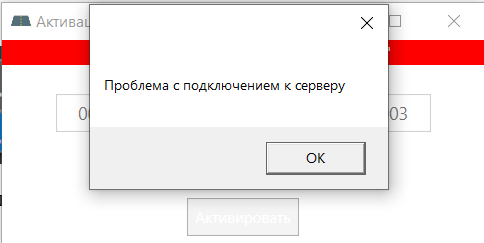


Рисунок 5.7 – Результат c выбором другого формата картинки

Дополнительно, была проведена проверка отправки уже активированного или использованного ключа, после отправки мы видим сообщение указывающее о том что ключ уже активирован или его не существует под полем ввода.

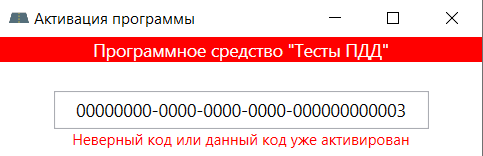


Рисунок 5.8 – Результат c выбором другого формата картинки

В итоге, процесс активации программного средства был полностью протестирован, следовательно, пользователь сможет без проблем активировать свое программное средство, данные тесты позволили убедиться в правильной функциональности системы активации продукта.

## 5.2 Вывод

В результате проведенного тестирования, было подтверждено успешное функционирование всех компонентов программного средства. Процесс активации протестирован и соответствуют заявленным функциональным требованиям. Полученные результаты тестирования подтверждают надежность и эффективность программного средства.

# 6 Руководство по установке и использованию

При первом запуске приложения будет открыто окно активации рисунок 6.1. с помощью данного окна можно активировать приложения при подключении компьютера к сети.

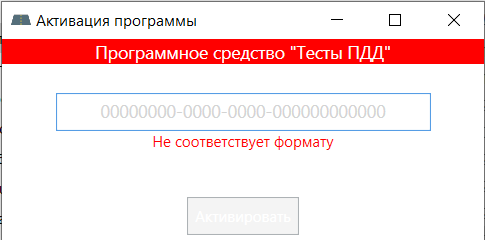


Рисунок 6.1 – Окно активации

После нажатия на кнопку “Активировать”, запуститься процесс активации и будет открыто основное окно рисунок 6.2, при следующих запусках приложение будет открываться уже сразу с основного окна.

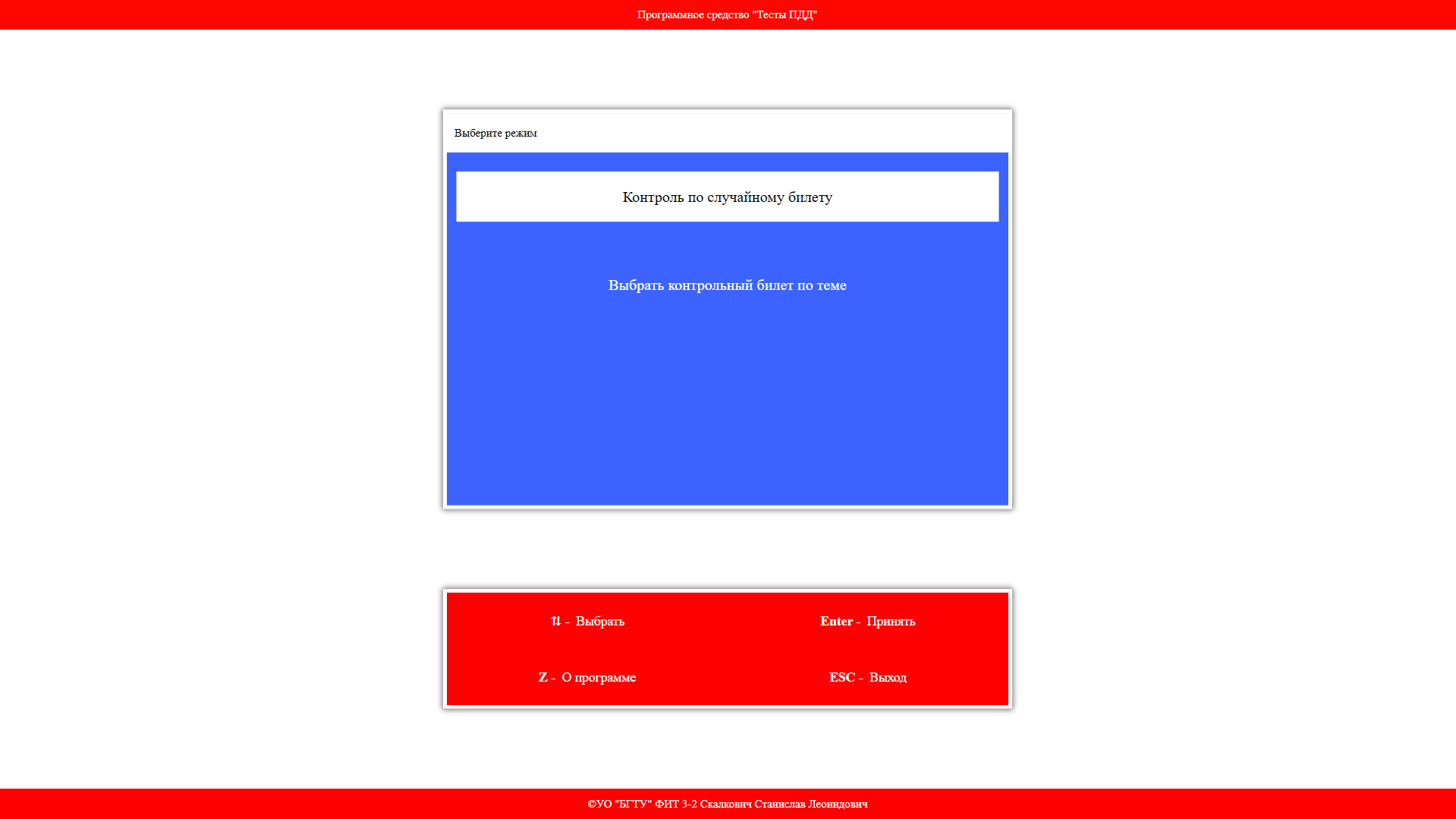


Рисунок 6.2 – Основное окно

С основной страницы пользователь может перейти на страницу справок окно рисунок 6.3 нажатием кнопки Z на клавиатуре,.

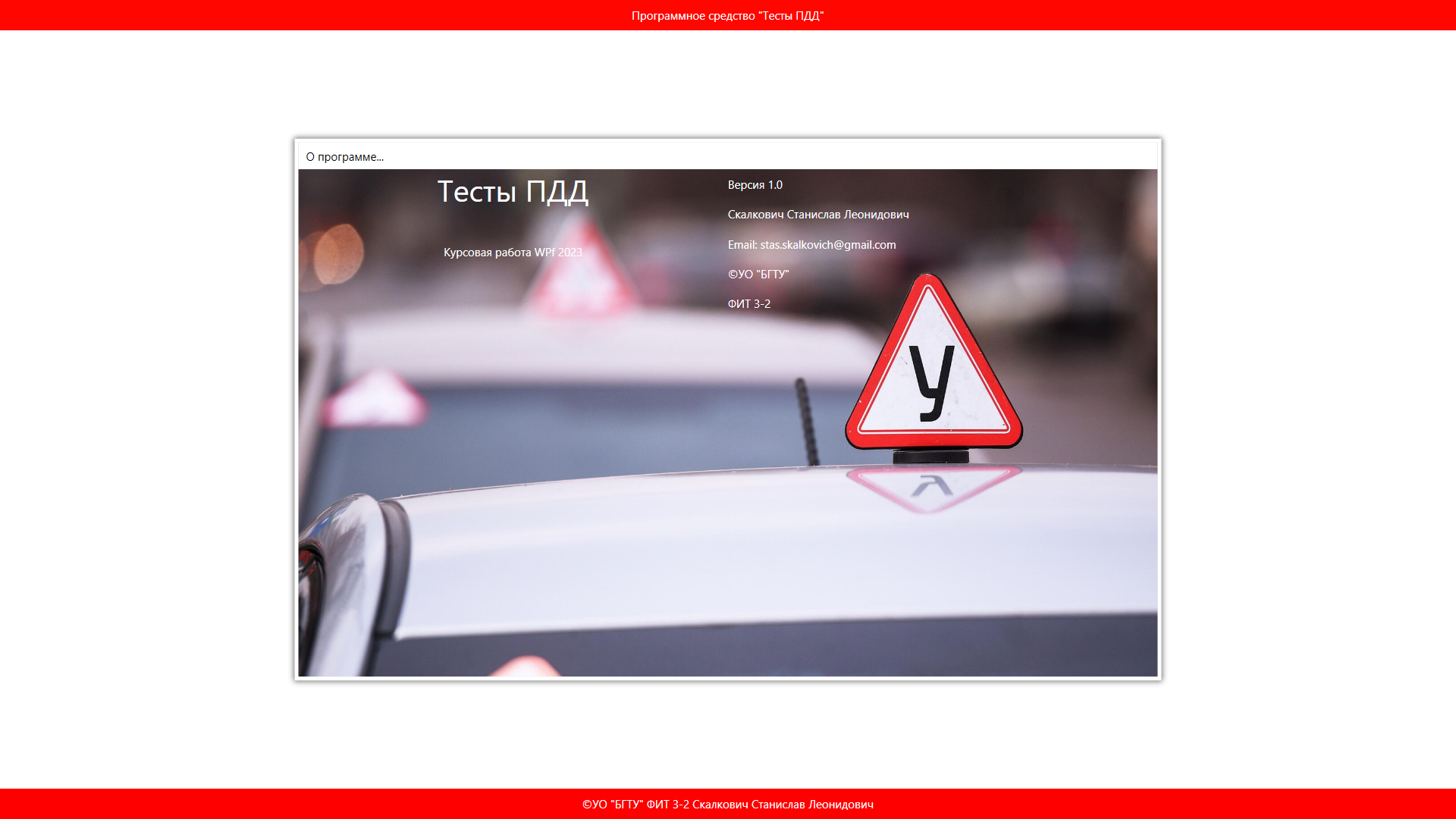


Рисунок 6.3 – Страница справок

С основной страницы пользователь может выбрать, нажатием на стрелочки вверх и вниз на клавиатуре, режим работы приложения. Нажатием кнопки Enter у пользователя запуститься тест окно рисунок 6.4.

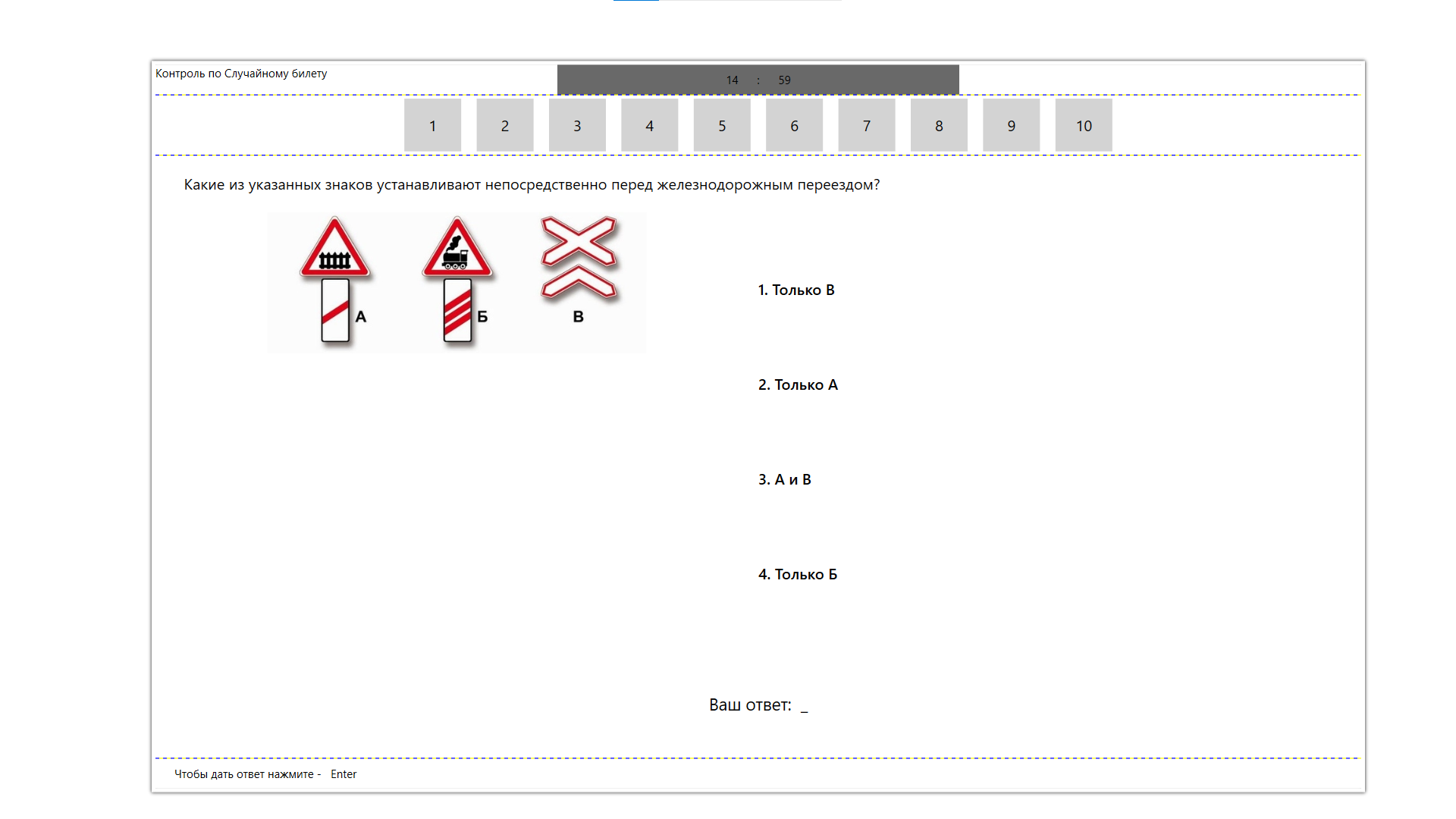


Рисунок 6.4 – Страница теста

На странице теста можно увидеть вопрос и варианты ответа, нажав на цифру 1\2\3\4 на клавиатуре пользователь выбирает вариант ответа, нажав backspace пользователь удалит свой выбор, по нажатию на Enter будет показано верный ли ответ, при последующем нажатии на Enter его перенесет на следующий вопрос или страницу с краткими результатами рисунок 6.5.



Рисунок 6.5 – Странциа кратких результатов

На странице кратких результатов пользователь может увидеть был ли пройден тест и количество верных и неверных ответов, по нажатию на Enter его перенесет на страницу результатов теста рисунок 6.6.

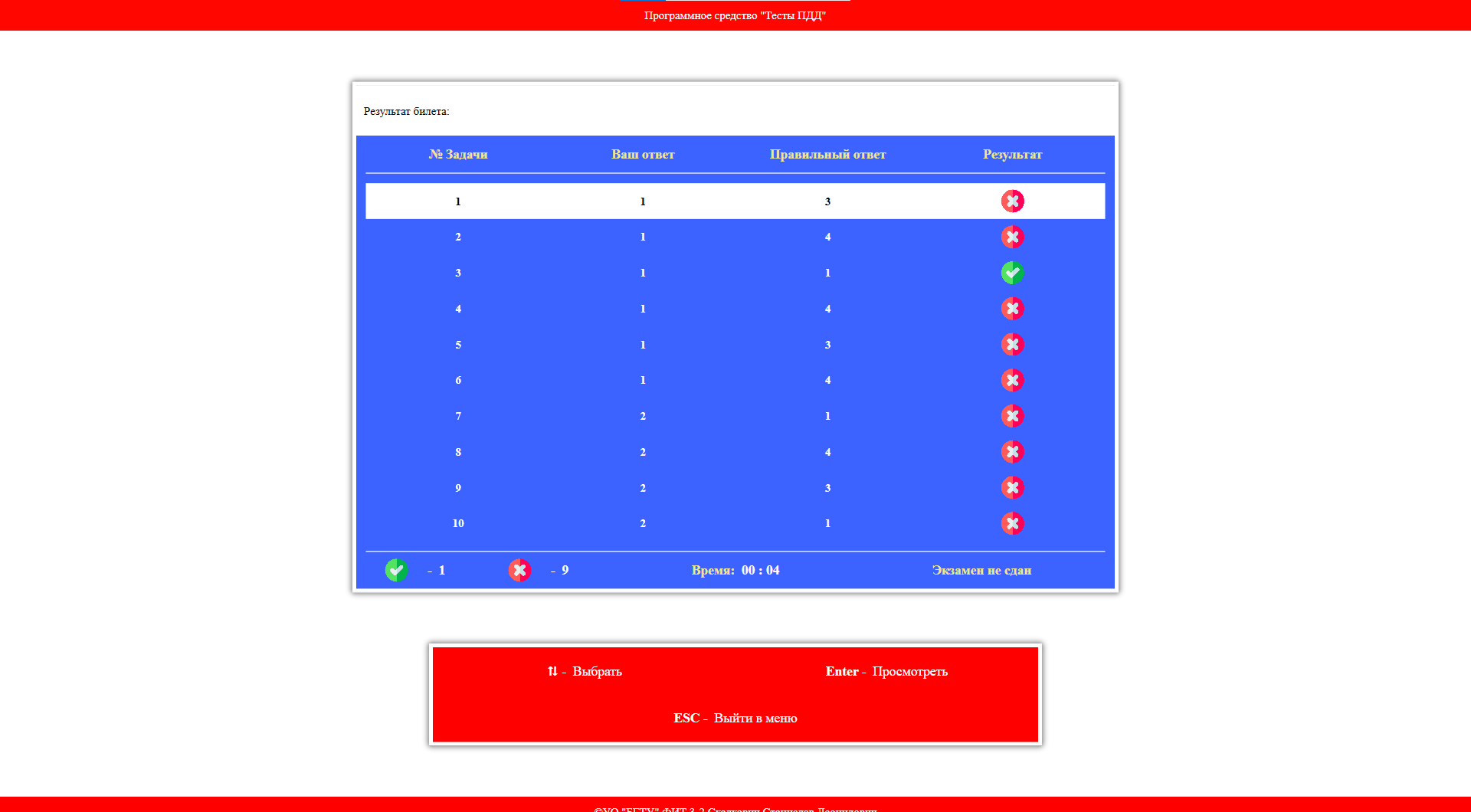


Рисунок 6.6 – Страница результатов

На странице результаты теста пользователь может используя стрелочки выбрать интересующий его вопрос и нажав Enter перейти на страницу работы над ошибками рисунок 6.7.

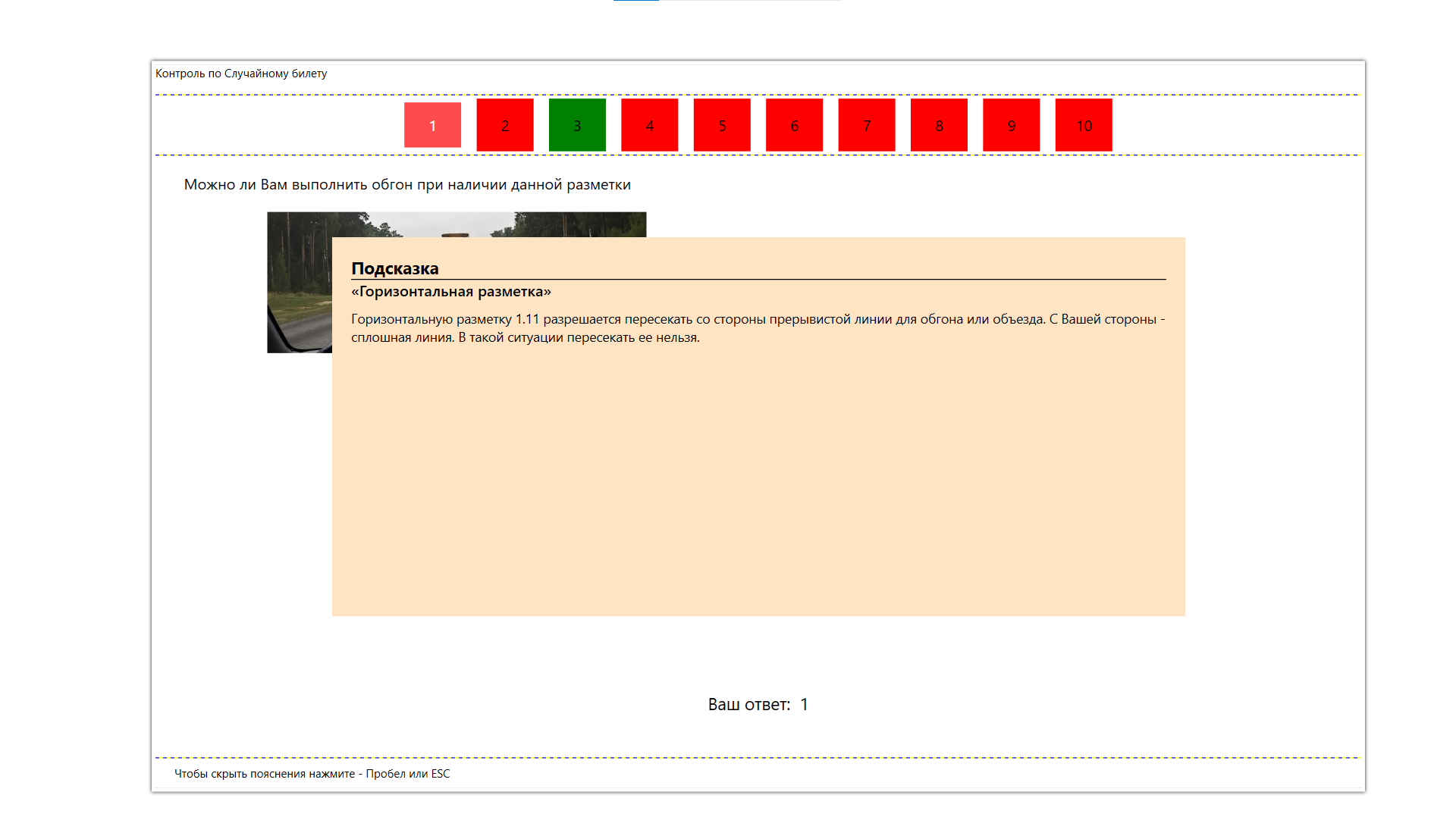


Рисунок 6.7 – Страница работы над ошибками

На странце работа над ошибками пользователь видит свой ответ и правильный, а также нажав пробел может увидеть описание вопроса. Чтобы вернуться в Основное Окно необходимо несколько раз нажать клавишу ESC. Для выхода из приложения необходимо нажать ESC на основной странице, после чего выбрать на открывшейся страницы выхода рисунок 6.8 выбрать да\нет и подтвердить выбор нажатием на Enter. После чего приложение закроется.

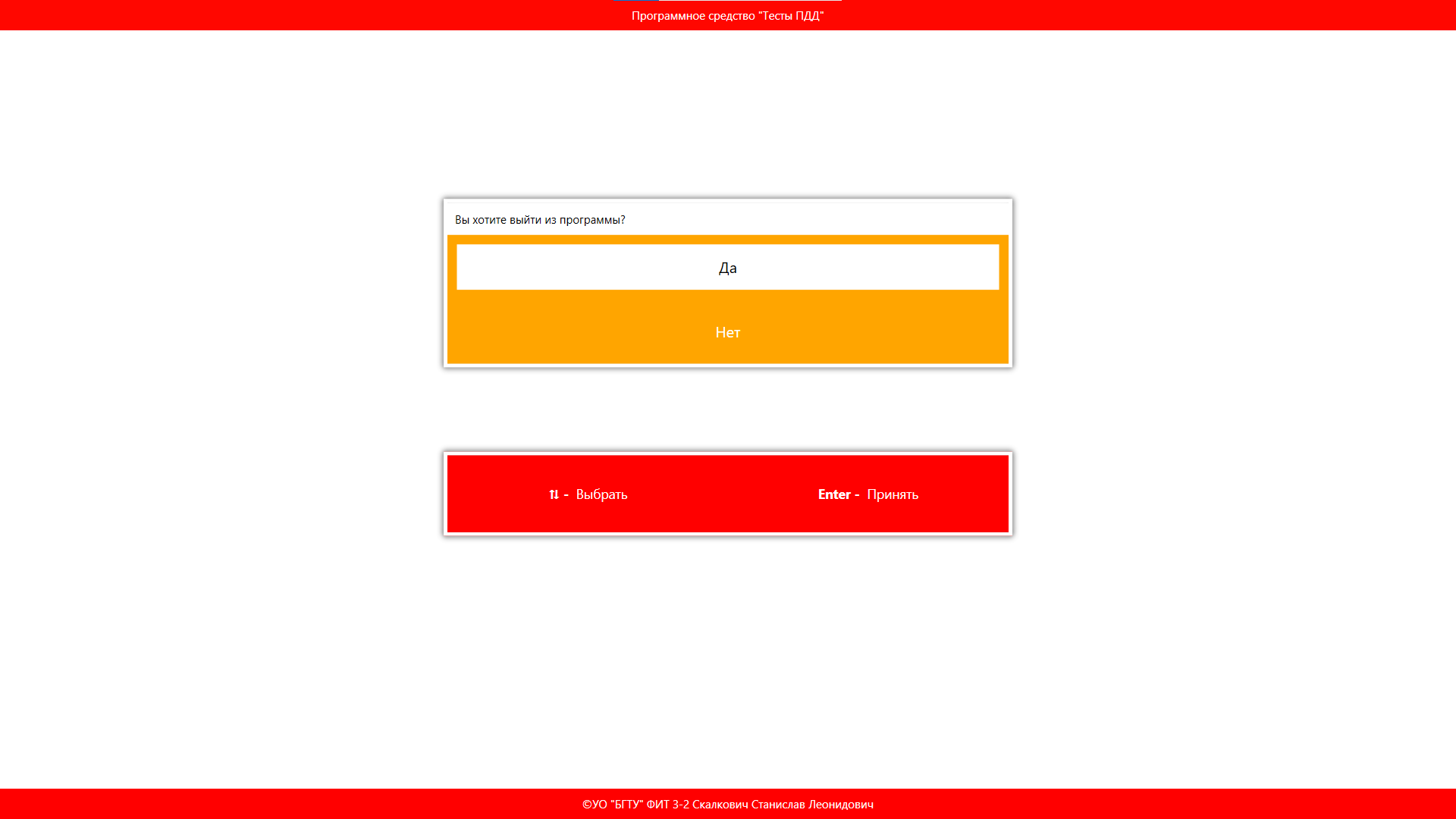


Рисунок 6.8 – Страница выхода

Програмное средство имеет простой мингималистичный дизайн и простую систему управления через клавиатуру, что позволяет создать персонализированный опыт и удобное взаимодействие с интерфейсом.

# Заключение

В ходе разработки программного средства “Тесты ПДД” были успешно реализованы его основные функциональности, предоставляя пользователям возможности прохождения тестов, ознакомление с результатами тестов, работа над ошибками. Особое внимание уделено персонализации пользовательского опыта через настройку взаимодействия с приложением исключительно через клавиатуру. Использование технологии Windows Presentation Foundation (WPF) и языка программирования C# обеспечивает современный и интуитивно понятный интерфейс.

При разработке програмного активно использовались современные технологии. В результате, приложение обеспечивает эффективное управление данными, высокую отзывчивость.

Для защиты данных использовался алгоритм шифрования RSA на основе приватного и публичного ключей. В программном средстве данные из БД расшифровываются приватным ключом при обращении к ним. Это обеспечивает защиту индивидуално составленых вопросов от кражи.

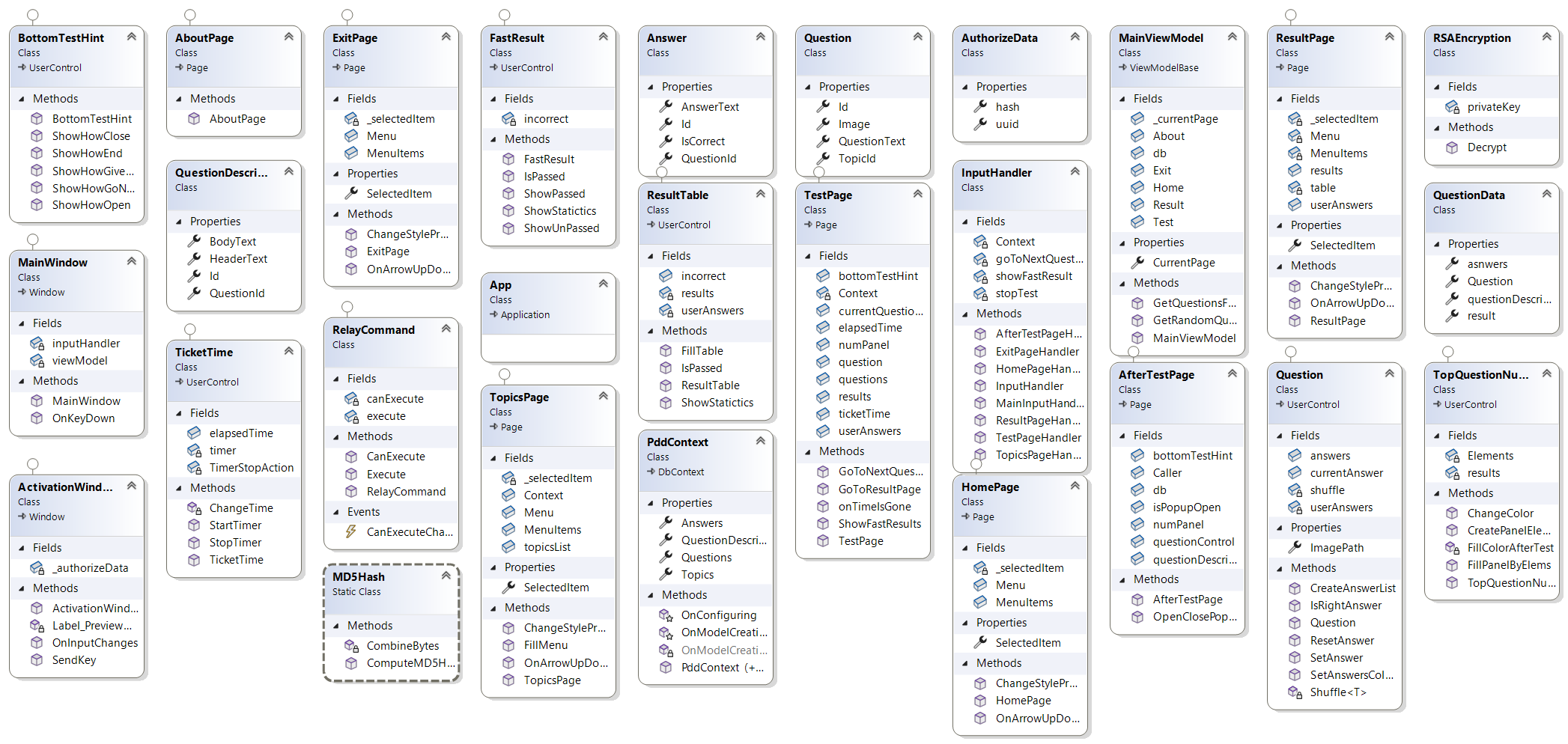
Программное средство готово полностью удовлетворять потребности пользователей “Прямо с коробки”, для начала его использования необходимо ввести ключ активации при подключении к глобальной сети интернет, после чего программным средством можно пользоваться автономно от сети, что может быть удобным не только для отдельных пользователей, но и для автошкол.

# Список использованных источников

1. dosaaf.net [Электронный ресурс] // Режим доступа: https://dosaaf.net/ – Дата доступа: 18.11.2023.
2. Adrive [Электронный ресурс] // Режим доступа: https://adrive.by/study/ – Дата доступа: 18.11.2023.
3. Руководство по SQLite // [Электронный ресурс] / Режим доступа: https://metanit.com/sql/sqlite/ – Дата доступа: 18.11.2023.
4. Microsoft Visual Studio [Электронный ресурс] – https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio – Дата доступа 12.12.2023.
5. Руководство по WPF // [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/wpf/ – Дата доступа: 12.12.2023.
6. Руководство по XAML // [Электронный ресурс]. – Режим доступа: https://www.tutorialspoint.com/xaml/index.htm– Дата доступа: 25. 12.12.2023.
7. Блинова, Е.А. Курс лекций по Базам данным / Е.А. Блинова. – Минск: БГТУ, 2019. – 175 с.

# Приложение А

Диаграмма классов



# Приложение Б

Диаграмма базы данных PDDTestBelarus

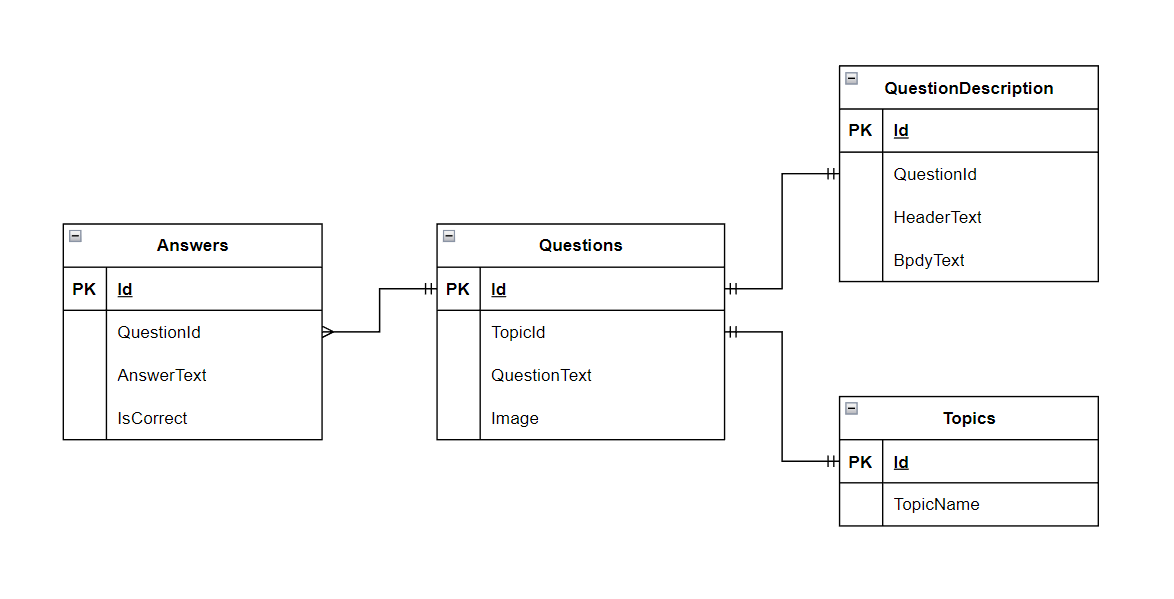
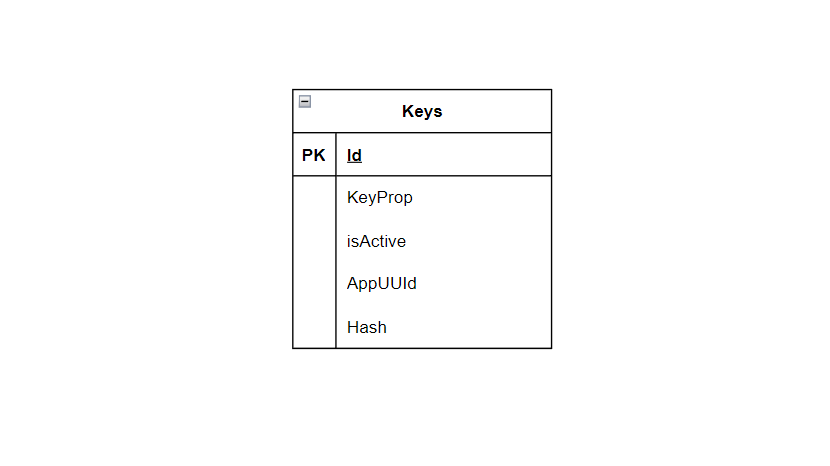
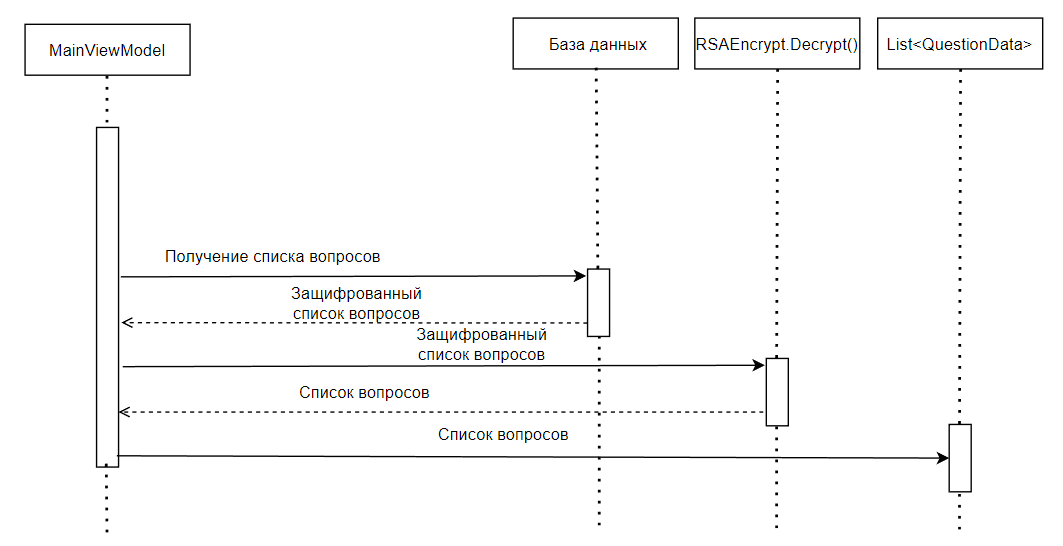


Диаграмма базы данных KeyServer



# Приложение В

Диаграмма последовательности



# Приложение Г

Use-case диаграмма

