Документация проекта «Статотчетность»

# 1. Введение

Проект «Статотчетность» представляет собой веб-сервис, предназначенный для приема, хранения и проверки статистических отчетов, предоставляемых филиалами в аппарат дирекции.   
Система разработана для упрощения взаимодействия между филиалами и центральным офисом, а также для ускорения проверки и обработки данных.

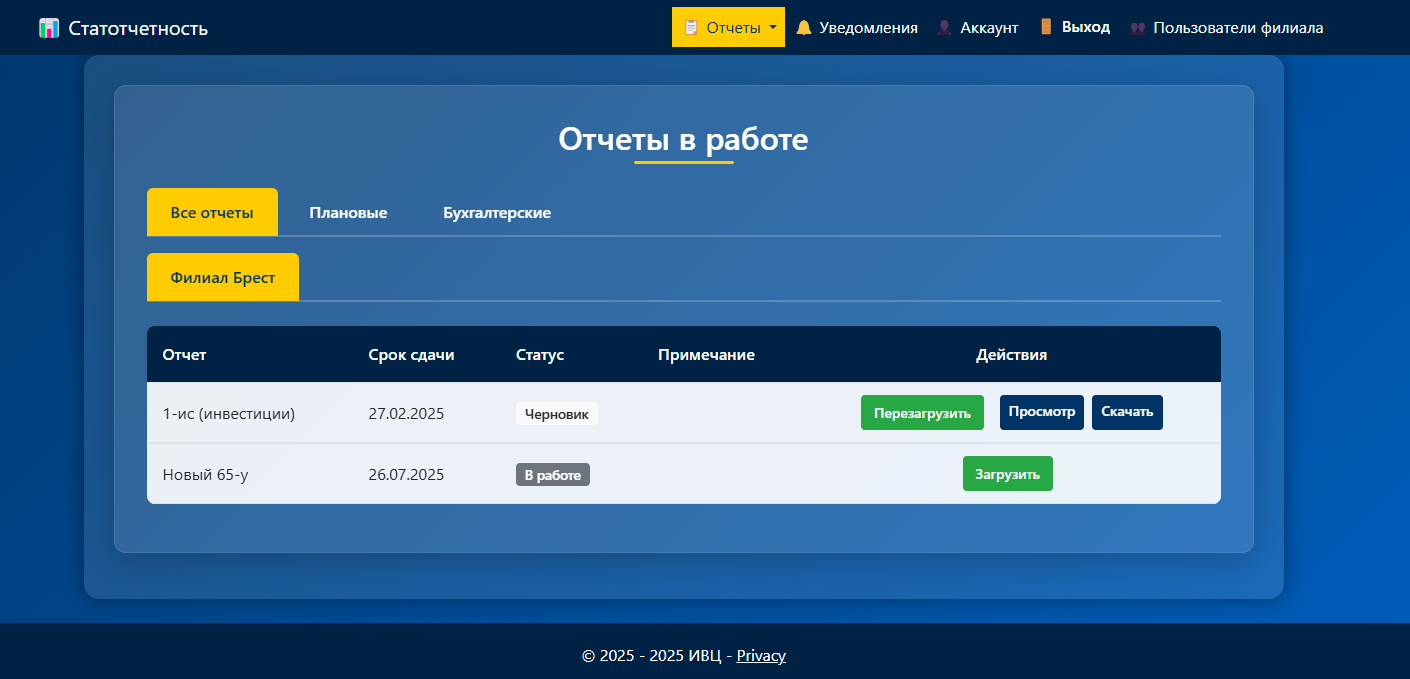
Основной сценарий использования: филиалы заполняют отчеты на внешнем портале «Электронный респондент», проходят контроль без подписи, и после успешной проверки загружают их в систему «Статотчетность». Далее отчеты проверяются сотрудниками аппарата дирекции (ПЭБ и ОБУиФ). В зависимости от результатов проверки, отчет либо принимается, либо возвращается на доработку с пометками для исправления.

***(филиал → портал → загрузка → проверка → статус → сводный отчет)***

# 2. Функциональные требования

## 2.1 Загрузка отчетов

Система позволяет загружать отчеты в формате Excel, предварительно прошедшие проверку на портале «Электронный респондент».   
После загрузки отчета в дирекции проверяются структура файла, наличие необходимых листов и корректность метаданных (тип отчета, период, филиал).

Для печати отчета необходимо его скачать  


***Скриншот формы загрузки отчета***

## 2.2 Управление статусами

Отчет в системе может находиться в одном из следующих статусов:  
- В работе — отчет ждет загрузки до финального срока сдачи;

- Черновик — загружен, но не проверен;  
- Необходима корректировка — обнаружены ошибки, требуется исправление;  
- Принято — отчет окончательно принят.

Переход между статусами осуществляется в зависимости от действий пользователя или проверяющего сотрудника.

***Скриншот интерфейса смены статуса***

## 2.3 Проверка отчетов

Проверка осуществляется сотрудниками ПЭБ и ОБУиФ. Система позволяет добавлять текстовые замечания к отчету и изменять его статус.  
При повторной загрузке исправленного отчета статус обнуляется, и он снова отправляется на проверку.

***Скриншот окна с примечаниями***

## 2.4 Сводные отчеты

Система позволяет формировать сводные отчеты на основе данных из филиалов.  
Сводный отчет может быть сгенерирован:  
- По всем филиалам;  
- По нескольким выбранным филиалам.

Процесс формирования включает парсинг структур, извлеченных из Excel, и их агрегацию по заданным правилам.

***Скриншот интерфейса генерации сводного отчета***

## 2.5 Уведомления

Система уведомляет пользователей о приближении или просрочке срока сдачи отчета. Уведомления отображаются в интерфейсе пользователя.  
Сроки привязаны к типу отчета и хранятся в таблице SubmissionDeadline.

***Скриншот уведомлений***

## 2.6 Учет пользователей

Используется двухуровневая авторизация:  
1. Аутентификация филиала по УНП и паролю;  
2. Вход пользователя филиала с личными учетными данными.

|  |
| --- |
| **Рисунок 1 – Авторизация филиала** |

|  |
| --- |
| **Рисунок 2 – Авторизация пользователя** |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

В системе предусмотрены роли: ответственный филиала, пользователь филиала, сотрудник ПЭБ, сотрудник ОБУиФ, администратор.

***Скриншоты страниц логина филиала и пользователя***

# 3. Архитектура системы

Система разработана с использованием архитектуры Onion (луковичной архитектуры), которая позволяет четко разделить уровни ответственности и упростить сопровождение и масштабирование проекта.  
Основные слои:  
- Core: содержит бизнес-логику, интерфейсы, DTO, сущности.  
- Infrastructure: реализация интерфейсов, работа с базой данных (EF Core), миграции.  
- Web (MVC): контроллеры, представления, сервисы авторизации и взаимодействие с пользователем.  
Система использует репозиторий-паттерн, асинхронные методы и строгую типизацию.

[Скриншот: Структура проекта в Visual Studio]

# 4. Технологии и стек

Проект разработан на платформе ASP.NET Core MVC с использованием следующих технологий:  
- ASP.NET Core MVC 8  
- Entity Framework Core  
- EPPlus (для работы с Excel-файлами)  
- MS SQL Server  
- HTML, CSS (Bootstrap, кастомные стили)  
- JavaScript (при необходимости взаимодействия с DOM)  
- Hosted Services (для автоматических проверок отчетов)

# 5. Структура базы данных

База данных включает в себя следующие сущности:  
- Report: отчеты, загружаемые филиалами  
- ReportTemplate: шаблоны отчетов  
- SubmissionDeadline: сроки сдачи отчетов  
- Branch: информация о филиалах  
- User: пользователи системы  
- ReportStatus: статусы отчетов (Черновик, Проверено, Необходима корректировка, Принято)  
- SummaryReport: сводные отчеты  
- ReportAccess: права доступа пользователей к отчетам

[Скриншот: Диаграмма сущностей или миграции EF]

# 6. Роли и права доступа

Система различает следующие типы пользователей:  
- Ответственные за филиалы: видят и управляют только своими отчетами.  
- ПЭБ и ОБУиФ аппарата дирекции: видят все отчеты всех филиалов, могут проверять, оставлять примечания, изменять статусы.  
- Администратор: управляет пользователями, филиалами, шаблонами отчетов, настройками доступа.

[Скриншот: Интерфейс прав доступа или таблица пользователей]

# 7. Сценарии использования

1. Ответственный филиала входит в систему, загружает проверенный отчет в формате Excel.  
2. ПЭБ или ОБУиФ проверяют отчет, меняют статус, при необходимости добавляют примечания.  
3. При необходимости вносятся корректировки и загружается новый файл.  
4. ПЭБ/ОБУиФ формируют сводный отчет по всем или выбранным филиалам.  
5. Система уведомляет о сроках сдачи отчетов.  
6. Администратор управляет структурами, пользователями, шаблонами.

[Скриншот: шаги сценария - загрузка, проверка, сводный отчет]

# 8. UI-интерфейсы

Система содержит следующие пользовательские интерфейсы:  
- Двухэтапная авторизация (филиал → пользователь)  
- Главная страница со списком отчетов и их статусами  
- Страница загрузки и замены отчетов

- Страница генерации сводного отчета  
- Интерфейс просмотра и проверки отчетов  
- Интерфейс администратора: управление филиалами, пользователями, шаблонами  
- Архив отчетов с фильтрами

[Скриншоты: интерфейсы каждой страницы]

# 9. Развертывание и настройки

Проект может быть развернут:  
- На Windows-сервере: с использованием IIS или kestrel + reverse proxy  
- На Linux: через Nginx + systemd сервис, используя kestrel  
Для деплоя требуется .NET SDK, SQL Server, настройки appsettings.json.  
Поддерживается локальный запуск и деплой в пределах корпоративной сети с использованием поддельного домена (через hosts).

# 10. Тестирование

Проект протестирован на следующих уровнях:  
- Юнит-тесты (проверка сервисов)  
- Интеграционные тесты (API и взаимодействие с БД)  
- UI-тесты (ручная проверка отображения и логики на страницах)  
- Ручное тестирование бизнес-процессов: сценарии от входа до формирования сводного отчета

# 11. Поддержка и сопровождение

В проекте предусмотрены механизмы логирования и уведомлений, что упрощает сопровождение.  
Журналирование действий пользователя и ошибок помогает отслеживать проблемы.  
Все бизнес-правила централизованы в сервисах, что упрощает поддержку и расширение логики.