ТЕХНИЧЕСКОЕ ЗАДАНИЕ

НА РАЗРАБОТКУ

МНОГОУРОВНЕВОЙ

ASP.NET MVC СИСТЕМЫ

ДЛЯ СТУДЕНЧЕСКОЙ БИБЛИОТЕКИ

|  |  |
| --- | --- |
| **Версия документа: 1.0** |  |
| **Дата создания:** 05.09.2018 |  |
| **Подготовлен:** 05.09.2018 | *Аюпов А.Х.* **группа: ПИ 1-14**  **E-mail: litpulla@mail.ru** |
|  |  |

# 1. Введение

## Аббревиатуры и обозначения

ASP (Active Service Page) – Фреймворк.

MVC (Model View Controller) – Паттерн.

N-Layer Архитектура – (Н-Слой) архитектура, в которой приложение разбивается на три логических уровня.

DAL (Data Access Layer) – уровень доступа к данным.

BLL (Business Logic Layer) – уровень бизнес логики.

WEB – здесь, уровень представления.

ПП – Программный Продукт.

ПО – Программное Обеспечение.

## Назначение

Данный документ содержит описание проекта, его предметную область и характеристики. Также содержит всю необходимую информацию о сроках и ответственных лицах, участвующих в разработке ПО.

## Предмет

Предметом данного документа является описание требований к web-ориентированной системе для библиотекарей, а также её особенности.

## Цель создания

Описание проблемы

На данный момент бизнес-процессы, происходящие в библиотеке не автоматизированы. Библиотекарям приходится хранить всю информацию о книгах и студентах в бумажной форме в больших стеллажах, порой выделяя для этих целей отдельную комнату. При единовременном приёме/выдачи большого количества книг создаётся коллапс, в виду невозможности ускорения процесса.

Целью создания данного ПП является автоматизация деятельности библиотекаря на стадии записи новых студентов, книг, а также выдачи и приёма книг. Данное приложение позволит «в пару кликов» получать и выдавать книги, а также исключит необходимость в хранении карточек в физическом виде.

# 2. Функциональные требования *2.1 Бизнес-требования*

Отсутствуют

### *2.2 Пользовательские требования*

* Возможность добавления новых студентов.
* Возможность добавления новых групп.
* Возможность добавления новых книг.
* Возможность выдачи необходимого количества книг студенту.
* Возможность приёма книг от студента.
* Возможность удаления студентов.

### *2.3 Практичность*

Система позволяет избежать бумажной волокиты, а также исключает возможность потери соответствующих бумаг.

**3. Функциональные требования**

### *3.1 Категории пользователей*

**Admin** – библиотекарь. Доступен весь функционал системы.

### *3.2 Производительность*

### Требований нет.

### *2.3 Безопасность*

Так как система не имеет выхода в интернет, требований к безопасности не предъявлялось.

### *2.4 Требования к данным*

Требований нет.

### *2.5 Требования к преобразованию данных*

Требований нет.

### *2.6 Требования к пользовательской документации*

Требуется наличие полной и подробной документации по всем функциям, включающую в себя скриншоты.

### *2.7 Требования к лицензированию*

### Не обязательно.

# 3. Функциональное описание решения

## 3.2 Интерфейсы

### *3.2.1 Пользовательский интерфейс*

Интерфейс должен быть интуитивно понятен и представлять из себя сайт.

### *3.2.2 Аппаратный интерфейс*

### *3.2.2.1 NET Framework*

Платформа Microsoft .NET Framework версии 4.0 или выше.  
**Для работы сайта обязателен режим работы пула приложения ASP.NET - integrated.**

### *3.2.2.2 Дисковое пространство*

Система занимает 500 Mб под хранение исполняемых файлов и требует минимум 1 Tб под базу данных.

### *3.2.3 Программный интерфейс*

Cервер должен работать под управлением одной из следующих операционных систем:  
     - Windows Server 2012 / 2012 R2  
     - Windows Server 2008 / 2008 R2

В системе должен быть установлен компонент IIS (Internet Information Services) версии 7 и выше.

Доступ к веб-сайту осуществляется с помощью одного из современных веб-браузеров (Opera, Internet Explorer, Google Chrome и прочее).

## 3.3 Обработка ошибок

Основные ошибки:

1. Ошибка 404 – файл не найден.
2. Ошибка 500 – внутренняя ошибка сервера.

Описанные выше ошибки обрабатываются системой и выдаются в представление через веб-интерфейс.

## 3.4 Источник данных

Основным источником данных является база данных, которая содержит информацию о студентах, книгах, группах и заёмах книг определенными студентами.

## 3.5 Преобразование данных

Все текстовые данные преобразуются в кодировке UTF-8.

## 3.6 Проектные требования и ограничения

Системой управления базой данных является MySQL.

Используются язык разметки HTML, каскадные таблицы стилей CSS, язык программирования JavaScript с использованием AJAX, PHP.

# Дополнение

В случае обнаружении необходимости разработчик имеет право вносить корректировки.