# Общие сведения

В рамках текущего задания требуется реализовать функционал (подпрограмму/модуль) для работы с нормативно справочной информацией (НСИ) в ВУЗе.

## Основные цели

Основными целями создания модуля НСИ («Справочники») являются:

* Унификация процесса работы за счет предоставления единого, удобного инструмента управления НСИ для пользователей системы
* Разграничение ответственности владения НСИ за счет разграничение прав доступа на основе ролевой модели.
* Минимизация трудозатрат линии поддержки по ведению НСИ
* Предоставление настраиваемого силами поддержки инструмента, позволяющего управлять НСИ (создавать, изменять, добавлять справочники)

## Место модуля в системе

Электронная информационно-образовательная среда (ЭИОС) вуза – это интегрированная совокупность информационным и образовательных ресурсов (в основном web приложений, систем, платформ, порталов и пр.) информационно-телекоммуникационных технологий и соответствующих технических и технологических средств.

НСИ в такой среде рассматривается как отдельная система с возможностью ее интеграции (использовании) в любую из систем ЭОИС находящейся под управлением (разработкой) отдела разработки программного обеспечения вуза.

На текущий момент в одной из систем, которая представляет большой монолит реализована НСИ, но это решение имеет ряд существенных недостатков:

* Решение практически невозможно в полном объеме повторно использовать в других системах ЭОИС, даже в рамках монолита повторное использование приводит к большому объему дублирования кода (в том числе и бизнес сценариев)
* Решение адаптировано только под работу с одним источником данных, а именно таблица базы данных MSSQL
* Решение плохо справляется с большими объемами данных в следствии чего UI имеет значительную задержку по отклику (в плоть до ошибки по времени ожидания), при этом кеширование данных в каком-то виде заложено

# Требования к системе

## Функциональные требования

Задача проектируемой системы – позволить пользователю выполнять следующие действия:

* Редактировать данные справочника: добавлять, изменять, удалять строки данных. При этом должны обеспечиваться следующие возможности:
  + контроль уникальности данных по заданному набору атрибутов (полей)
  + поддержка следующих типов данных: строковый, числовой, логический, дата и время, справочник, изображение
  + валидация полей ввода на обязательность заполнения и по формату данных
  + контроль прав редактирования по уровню доступа пользователя
* Просматривать данные справочника в виде табличной сетки в отдельном модальном окне, при этом пользователь должен иметь возможность: быстрого поиска и расширенной фильтрации данных, сортировки, постраничного просмотра данных. Сегментация данных (предварительная фильтрация) должна осуществляться по параметрам доступа пользователя
* Фильтровать данные справочника с использованием расширенного фильтра по заранее настроенным полям фильтрации
* Выполнять вывод отфильтрованных данных на печать в виде таблицы
* Выполнять выбор значений (строк) из справочника, допускается множественный выбор значений. Выбор значений выполняется при участии справочника в составе атрибутов (полей) другого справочника или при размещении справочника на форме web-приложения в виде универсального элемента управления. Выбранные значения должны отображаться в виде отдельного списка в окне справочника с возможностью их удаления.
* Выполнять настройку справочников, через интерфейс администратора (конструктор справочников): создание, редактирование, удаление справочников. Настройка представляет процесс заполнения метаинформации по справочнику (описание источника данных, параметров справочника, доступа, полей, фильтров)

## Нефункциональные требования

При проектировании и реализации системы должны быть учтены следующие требования:

* Выполнять запись в журнал истории основные действия пользователя над данными справочника (добавление, изменение, удаление) в расширенном формате с возможностью дальнейшей агрегации и анализа журнала
* Реализовать возможность кеширования данных справочника, а также кешировать метаинформацию справочника
* Реализовать универсальный UI (web элемент управления) для возможности его повторного использования в других web-приложениях ЭОИС
* Реализовать безопасный доступ к Web API c использованием токена безопасности (открытого и закрытого ключа) или сессионных данных пользователя
* Реализовать гибкую возможность добавления новых источников данных справочника таких как: различны БД, xml-файлы, web-службы (REST API, SOAP) и пр.
* Обеспечить высокую скорость отклика, вывода данных справочника на экран (в пределах 3 сек.)

# Вид приложения

Данное решение должно быть представлено двумя web-приложениями:

1. Первое приложение представляет настроечную часть для администратора, отвечает за создание и ведение справочников (создание, редактирование, удаление). Приложение может быть реализовано в виде web-приложения MPA на базе паттерна MVC с использование Repository в качестве паттерна доступа к данным. Данный выбор обоснован тем что к части администрирования предъявляются более мягкие требования, при этом MVC выбран как классический подход в таких приложениях. Паттерн Repository выбран как один из стандартных и удобных шаблонов доступа к данным, не добавляющим сложности.
2. Второе приложение отвечает непосредственно за наполнения справочников данными. Предполагается реализация в виде web-приложения SPA, с использованием паттерна MVC в его упрощенном виде WEB API с использование Repository в качестве паттерна доступа к данным. Часть представления данных должна быть реализована в виде универсального элемента управления с применением паттерна MVVM. Данный выбор обоснован тем что для реализации независимого элемента управления не должно быть его жесткой связи с целевым приложение, а также взаимодействие с компонентом не должно приводить к полной перерисовки страницы. Паттерн Repository выбран с целью абстрагирования от источника данных для справочника.