

## СОДЕРЖАНИЕ

1	АНАЛИТИЧЕСКИЙ ОБЗОР ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ УЧЕБНЫХ ЦЕНТРОВ . . . . .	6
1.1	ИТ-управление учебным центром: цели, задачи и функции . . . . .	6
1.2	Типы сайтов и обзор средств разработки сайтов . . . . .	7
1.3	Анализ систем управления контентом . . . . .	7
1.4	Паттерны и фреймворки . . . . .	7
1.5	Обзор Web-ресурсов для учебных центров . . . . .	7
2	АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ РАЗРАБОТКИ WEB-РЕСУРСА . . . . .	8
2.1	Обзор предметной области . . . . .	8
2.2	Современные технологий и подходы в разработке web-ресурсов . . . . .	8
2.2.1	Системы управления контентом . . . . .	8
2.2.2	Применяемые подходы к организации веб-ресурса . . . . .	9
2.2.3	Применяемые языки программирования . . . . .	11
2.2.4	Технологии разработки для front-end . . . . .	12
2.2.5	Технологии разработки для back-end . . . . .	13
2.2.6	Операционные системы . . . . .	14
2.2.7	Среды разработки и редакторы кода . . . . .	14
2.2.8	Редакторы диаграмм . . . . .	15
2.3	Определение требований к web-ресурсам . . . . .	15
2.4	Проверка соответствия существующей системы современным требованиям и технологиям . . . . .	16
2.4.1	Обзор версий оформления страниц сайта и их развития . . . . .	16
2.5	Постановка требований к разрабатываемому web-ресурсу . . . . .	16
2.6	Постановки задачи . . . . .	17

# **1 АНАЛИТИЧЕСКИЙ ОБЗОР ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ УЧЕБНЫХ ЦЕНТРОВ**

## **1.1 ИТ-управление учебным центром: цели, задачи и функции**

По части вопроса управления учебным центром можно выделить несколько пунктов, действующих в себе полностью или частично отрасль информационных технологий и представляющий интерес в данной работе. Итак, соответствующие темам вопросы:

- Управленческие - от поддержания внутренней базы кадров до бухгалтерского учета;
- Технической поддержки - хранение информации об используемой технике, ПО и его версиях, параметрах настройки, а также распределении по кабинетам/аудиториям;
- Информационно-оповещательные - распространение информации о деятельности учебного центра и способах взаимодействия с ним;
- Внутреннего учета - обеспечение возможности получения, обработки и хранения данных о клиентах учебного центра (студенты, ответственные за них лица);
- Проверки выполнения заданий - предоставление функционала по сбору, выдаче и контролю за правильностью выполнения поставленных студентам задач;
- Контрольная - проверка доступов и способов взаимодействия внутри и между систем.

Данный список содержит пункты, которые могут пересекаться, быть вложенными друг в друга или взаимодействовать иным образом в зависимости от конкретных задач, поставленных перед учебным центром, специфике его целевой аудитории и прочих факторов.

Исходя из приведенных выше вопросов уже можно сделать вывод об индивидуальности подхода в проектировании и использовании ИТ-инфраструктуры. Теперь можно перейти к рассмотрению основной задачи данного раздела - определению ключевых требований к ИТ-управлению учебным центром.

### Цели

Наверное, самым простым вопросом относительно применения ИТ в управлении учебным центром является постановка цели. В данном случае лучшей формулировкой будет - оптимизация процессов взаимодействия сотрудников и технического обеспечения учреждения для получения наиболее продуктивного подхода в организации учебного процесса и, как результат, максимизация прибыли.

### Задачи

В этом пункте надо понимать постановку вопроса и конкретные требования от заказчика - непосредственно организации. Но, тем не менее, можно сформулировать общие пункты-задачи для ИТ-инфраструктуры учебного центра:

- Оптимизация способов взаимодействия между действующими лицами относительно услуг учебного центра (клиент - преподаватель, преподаватель - студент, студент - учебная система и т.д.);
- Предоставление информации о деятельности учебного центра (состав учебных программ, наличие свободных мест в группах, данные об учебном процессе и прочее);
- Формирование технической возможности автоматизированной работы студента с системой учреждения (проверка, выдача задач, ведение статистики);
- Обеспечение документооборота внутри организации (бухгалтерия, данные сотрудников и учеников);
- Ведение учета посещаемости и оценок за выполнение задач.

И снова считаю важным указать, что данный набор пунктов не является абсолютным и напрямую зависит от требований заказчика ИТ-системы.

### Функции

С данным пунктом проще всего - он должен содержать выжимку из приведенных выше, оформленную в четко сформулированные понятия. Итак, используя разобранные ранее вопросы можно выделить следующие функции ИТ-управления учебным центром:

- Цифровизационная - частичный или полный перенос бумажного документооборота в цифровой формат;
- Оптимизационная - модернизация имеющихся процессов, исключение лишних, преобразование неоптимальных;
- Социальная - расширение охвата целевой аудитории учреждения, привлечение новых клиентов>;
- Управленческая - предоставление контроля над элементами системы учебного центра;
- Статистическая - сбор и обработка статистики о процессах работы учреждения;
- Автоматизационная - перенос части вручную выполняемых задач на автоматически исполняемые.

Частичное или полное соответствие разрабатываемой ИТ-системы приведенным функциям может свидетельствовать о ее пригодности применения в деятельности учебного центра.

## 1.2 Типы сайтов и обзор средств разработки сайтов

w

## 1.3 Анализ систем управления контентом

е

#### 1.4 Паттерны и фреймворки

г

#### 1.5 Обзор Web-ресурсов для учебных центров

т

## **2 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ РАЗРАБОТКИ WEB-РЕСУРСА**

### 2.1 Обзор предметной области

Разработка веб-ресурса как предметная область - весьма современная и востребованная тема. На данный момент сложно представить компанию без веб-ресурса. При этом даже можно не указывать область деятельности компании – все виды коммерческого взаимодействия подразумевают контакт с внешним миром через средства связи и одно из наиболее распространенных и востребованных – веб-ресурс.

Само понятие веб-ресурса не конкретизирует форму представления, а лишь включает в себе способ передачи информации целевой аудитории. Таким образом под такое определение может попасть как полноценное приложение, так и, например, одностраничный сайт с картинкой-заглушкой.

### 2.2 Современные технологий и подходы в разработке web-ресурсов

#### 2.2.1 Системы управления контентом

Широкое распространение в последние годы получает направление упрощенного формирования веб-ресурсов. Простота подхода заключается в используемых средствах для достижения финального результата. Одним из таких средств являются системы управления контентом (Content Management System), заключающие в себе программное обеспечение, строящее процесс взаимодействия с пользователем по типу работы с конструктором. Основные выгоды, получаемые при выборе данного подхода сосредоточены вокруг использования уже готовых решений и их кастомизации под конкретные нужды заказчика. Процесс доработки при этом зачастую является менее требовательным к уровню подготовки исполнителя, а потому становится проще и доступнее даже для людей, не владеющих навыками взаимодействия с кодом.

За доступность в использовании приходится расплачиваться ограничениями, заложенными в саму платформу-конструктор. Зачастую эти ограничения выливаются в виде невозможности тонкой настройки отдельных компонентов системы под конкретные нужды пользователя. Эта проблема решается, обычно, несколькими путями - либо переходом на более кастомизируемый и подходящий под задачи конструктор, либо взаимодействием с разработчиком (зачастую на платной основе) с целью получения желаемых изменений в продукте. Второй вариант может проявляться в разных видах взаимодействия - единоразовое приобретение прав на самостоятельное внесение изменений, долгосрочный контракт с вендором. Еще одним из минусов такого подхода является вынужденная зависимость от предоставляемого ПО, а следовательно и от его разработчика.

Но даже с учетом приведённых недостатков, данный вид формирования веб-ресурса развивается. Вот лишь некоторые из самых популярных представителей современных систем управления контентом по версии портала CMS magazine []:

- Wordpress
- 1С-Битрикс

- Joomla
- Drupal
- Tilda Publishing

## 2.2.2 Применяемые подходы к организации веб-ресурса

Альтернативным подходом к использованию систем управления контентом, является изначальная разработка всей системы. Данный подход вовсе не отрицает возможности использовать наемный труд со стороны, а лишь указывается, что программные средства будут разработаны под индивидуальные требования заказчика.

С точки зрения подходов в работе веб-ресурсов можно выделить два направления:

- Многостраничное приложение или MPA - multi page application. Заключается в построении схемы обмена данными между клиентом и сервером, основанной на передаче цельных HTML-файлов. Данный способ приводит к перезагрузке страницы при каждом запросе;
- Одностраничное приложение или SPA - single page application. Заключается в ином подходе - подгрузке основной части приложения при инициализирующем запросе и дальнейшее наполнение данными при последующих запросах к серверу.

Данные подходы уже хорошо себя зарекомендовали и применяются повсеместно. Но поверх них существуют и другие уровни реализации приложения - непосредственное воплощение в технологиях.

Подходы в использовании определенного набора технологий в профессиональной среде называются "стек" (от английского stack - куча). Само название стека определяется его содержимым - набором инструментов, обычно отвечающим следующим задачам:

- Формирование клиентской части приложения - то что видит и с чем взаимодействует пользователь;
- Взаимодействие сторон клиента и сервера - построение каналов обработки сообщений между частями приложения;
- Хранение и взаимодействие с базой - собственно организация накопления, обработки и выдачи данных для использования приложения.

Например стек MEAN берёт своё название от имен применяемых технологий, решающих приведенные выше задачи:

- MongoDB - СУБД;
- Express.js - построение маршрутизации в приложении;
- Angular.js - клиентская (front-end) часть приложения;
- Node.js - серверная (back-end) часть приложения.

И простым изменением одного из компонентов можно получить новый стек - MERN. В данном случае за клиентскую часть будет отвечать другая технология - React.js.

Ещё одним важным этапом является выбор способа взаимодействия между клиентской и серверной частью. За данный пункт отвечают специальные интерфейсы прикладных программ - API (application programming interface), выполняющие роль посредника между частями приложения и определяющие сам способ передачи информации, её вид. Наиболее популярным является архитектурный подход REST, основывающийся на протоколе HTTP для транспортировки данных. Например, в приведённом выше стеке MERN/MEAN подход REST реализуется на базе технологии Express.js. Более подробно о строении подхода REST будет рассказано в

Альтернативным к REST подходом, является GraphQL, рассматривающий взаимодействие и работу с данными через графы. Данный подход также будет подробнее рассмотрен позже.

### 2.2.3 Применяемые языки программирования



Разработка веб-ресурсов вот уже несколько десятилетий вбирает в себя самые разные технологии и подходы. В данном разделе будут приведены наиболее востребованные и распространенные языки программирования и их сочетания, применяемые в индустрии на 2021-2022 год.

- HTML
- CSS
- PHP
- Ruby
- Python
- Java
- JavaScript

В целом приведенный список можно разделить на две важные части - языки для front-end части и back-end. Здесь важно указать, что подразумевают два этих понятия.

Front-end - часть веб-приложения, отвечающая в большей степени за визуальную составляющую - то, с чем взаимодействует пользователь. К чисто фронтovým технологиям можно отнести HTML и CSS - они отвечают именно за часть отображения и оформления контента.

Back-end - часть веб-приложения, отвечающая за обработку системной и бизнес-логики для дальнейшей её передачи в некотором виде пользователю через фронтową часть. К данной категории можно отнести языки, сосредотачивающие контроль над серверной стороной - Ruby, Java.

Хоть приведенные примеры и несколько категоричны, важно указать, что границы между четким front и back приложения и относящимся к ним языкам программирования имеют тенденцию размываться, что приводит к неоднозначности, например, такого утверждения: "JavaScript - чисто front-end язык программирования", которое может быть верным и неверными в зависимости от множества технических факторов.

Исходя из описанных выше данных, будут выбраны в предпочтение для реализации технологии, касающиеся языка JavaScript.

#### 2.2.4 Технологии разработки для front-end

В данном пункте приведены различные реализации надстроек над основным языком, называемые фреймворками. Фреймворк — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта []

Сам фреймворк играет ключевую роль в способе представления информации и взаимодействия с ней. Поскольку фреймворк является лишь расширением, ему требуется основание для работы. Таким основанием может являться один из языков программирования, который ложится в основу проработки внутренней логики и строения приложения. Наиболее популярные фреймворки, расширяющие возможности JavaScript:

- react
- angular
- vue

Дополнительно стоит упомянуть возможность работы с ”чистым” JavaScript - без применения фреймворков. Для такой ситуации можно применить подход описания систем управления контентом, представленные ранее в данной работе. Минусы и плюсы использования фреймворков также заключаются в ограничениях свобод разработчика, его зависимости от используемой технологии. Есть компании, определяющие подход своей разработки с приложением через применение чистого JavaScript без дополнений, но, как правило вместе с таким подходом повышается уровень требований к программистам ввиду необходимости реализации множества компонентов приложения.

### 2.2.5 Технологии разработки для back-end

Некоторая уникальность JavaScript заключается в возможности его применения как для front-end части, так и для back-end. Данная возможность появилась вместе с созданием Райаном Далом технологии Node.js в 2009 году.

Node.js - программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS или Electron для Linux, Windows и macOS) и даже программировать микроконтроллеры (например, tessel, low.js и espruino). В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом.

Сами технологии, основанные на Node.js многогранны, но наиболее востребованными для разработки веб-приложения являются:

- electron.js
- express.js

### 2.2.6 Операционные системы

Пункт, связанный с операционными системами, на мой взгляд, также является достойным обсуждения. С точки зрения веб-разработки нет большой разницы в применяемой платформе - основной набор технологий будет работать идентично на любой ОС.

- windows

- mac os
- linux

### 2.2.7 Среда разработки и редакторы кода

Одной из важных частей разработки является непосредственное взаимодействие с кодом - его редактирование, написание, удаление, чтение. Эти процессы являются критическими для разработчика, поскольку являются

- VS Code
- sublime
- atom
- VS

### 2.2.8 Редакторы диаграмм

Необходимы для составления диаграм взаимодействия Одним из наиболее популярных средств является универсальный язык моделирования UML [?]. Видов программного обеспечения, использующего данный язык моделирования много:

- Rational Rose
- Dia <http://dia-installer.de/>
- PlantUML <https://plantuml.com/ru/> Можно использовать для отображения объектов и сущностей формата JSON (т.к. используется ЯП javascript) - <https://plantuml.com/ru/json>  
Использование с латехом <https://plantuml.com/ru/latex>

## 2.3 Определение требований к web-ресурсам

Консорциум Всемирной паутины (англ. World Wide Web Consortium, W3C) — организация, разрабатывающая и внедряющая технологические стандарты

для Всемирной паутины. Основателем и главой Консорциума является сэр Тимоти Джон Бернерс-Ли, автор множества разработок в области информационных технологий. По состоянию на 29 мая 2019 года Консорциум насчитывает 444 члена.

W3C разрабатывает для Интернета единые принципы и стандарты (называемые «рекомендациями», англ. W3C Recommendations), которые затем внедряются производителями программ и оборудования. Таким образом достигается совместимость между программными продуктами и аппаратурой различных компаний, что делает Всемирную сеть более совершенной, универсальной и удобной.

Миссия W3C: «Полностью раскрыть потенциал Всемирной паутины путём создания протоколов и принципов, гарантирующих долгосрочное развитие Сети». Более конкретная цель W3C — помочь компьютерным программам достичь способности к взаимодействию в Сети (т. н. «сетевая интероперабельность», англ. Web interoperability). Применение единых стандартов в Сети — это ключевой шаг для достижения такого взаимодействия.

Две другие важнейшие задачи Консорциума — обеспечить полную «интернационализацию Сети» и сделать Сеть доступной для людей с ограниченными возможностями. Для решения первой задачи Консорциум активно сотрудничает с организацией «Юникод» (англ. Unicode) и рядом других рабочих групп, занимающихся международным сотрудничеством в Интернете и языковыми технологиями. Для решения второй задачи Консорциум не только сотрудничает с организациями соответствующего профиля, но и разработал свои собственные рекомендации, которые сейчас активно набирают популярность. Существует «Руководство по обеспечению доступности веб-контента (WCAG)», разработанное Консорциумом всемирной паутины (W3C). В нем четко расписаны все требования по контенту сайтов и его форматированию, чтобы максимум людей могли комфортно пользоваться информацией.

## 2.4 Проверка соответствия существующей системы современным требованиям и технологиям

### 2.4.1 Обзор версий оформления страниц сайта и их развития

<https://web.archive.org/web/20120219134344/http://www.meson.ru/>

## 2.5 Постановка требований к разрабатываемому web-ресурсу

## 2.6 Постановки задачи

Разработка веб-приложения в соответствии с представленными ранее подходами и технологиями.