# Введение

В данном курсовом проекте необходимо разработать сайт «Electronic Scheme».

Сайт — это совокупность логически связанных между собой веб-страниц. Обычно сайт в интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователем как единое целое.

В настоящее время существует много различных аналогов, например, EasyEDA, Circuit Sims, EveryCircuit и т.п. Данный курсовой проект отличается от других тем, что каждая схема находятся в открытом доступе и любой пользователь может просмотреть и прокомментировать ее. Поэтому данный проект похож больше на такие сайты как Habrahabr или Пикабу, только вместо обычных постов можно добавить электронную схему.

Основной целью выполнения курсового проекта является расширение, углубление знаний в области алгоритмизации и программирования на языке C# и в среде разработки Visual Studio 2015, расширение знаний о ASP .Net MVC и разработке сайтов, формирование навыков научно-исследовательской деятельности.

Задачи выполнения курсового проекта:

* систематизация научных знаний в области алгоритмизации и программирования на языке C#;
* изучение технологии ASP .Net MVC;
* углубление уровня и расширение объема профессионально значимых знаний, умений и навыков работы в среде быстрой разработки Visual Studio;
* формирование умений и навыков самостоятельной организации научно-исследовательской работы;
* овладение современными методами поиска, обработки и использования информации с помощью баз данных.

Данная пояснительная записка содержит следующие разделы:

1. Раздел «Постановка задачи» содержит формулировку задачи с указанием всех функций проекта, определение круга задач, которые должны быть автоматизированы в данном курсовом проекте, описание входных и выходных данных проекта.
2. Раздел «Вычислительная система» содержит обоснование выбора языка программирования и среды разработки проекта, описание операционной системы, в которой была реализована программа. Требования конфигурации программного и аппаратного обеспечения.
3. Раздел «Проектирование» содержит описание внешнего пользовательского интерфейса, структуру меню, алгоритма, функционально независимых подзадач и путей их решения.
4. В разделе «Описание программы» содержится описание всех модулей, функций, входящих в проект, их назначение и взаимодействие, особенности интерфейса программы.
5. В разделе «Отладка и испытание программы» описывается проверка работы программы на различных тестах и анализ полученных результатов.
6. Раздел «Описание и применение программы» содержит сведения о назначении программного средства, области применения, требования к необходимым программным и техническим ресурсам.
7. В «Заключении» описывается краткая формулировка проблемы и пути ее решения, использованные методы и средства, возможность дальнейшей модификации проекта.

# 1 Постановка задачи

## 1.1 Формулировка задания

Необходимо разработать сайт «Electronic Scheme». Реализовать возможность создавать и просматривать электронные схемы с описанием.

В проекте реализовать возможность регистрации и авторизации. Так же должна быть возможность авторизоваться через социальные сети (Вконтакте, Facebook, Tweeter).

После авторизации пользователю присваивается роль: «User» или «Admin».

Неавторизированный пользователь должен иметь возможность только просматривать, осуществлять поиск и фильтрацию контента.

User может делать все что и неавторизированный пользователь, а так же возможность: добавлять, удалять и редактировать свои посты, комментировать и оценивать свои и чужие посты. Каждый user имеет свою личную страницу, где он может поменять свое фото и увидеть все свои посты.

Admin имеет те же возможности, что и user, плюс к этому он может просмотреть список пользователей и лишить прав пользователя, оставив возможность только просматривать контент или дать права администратора. Так же admin может удалять и редактировать любые посты.

На сайте должна быть возможность поиска контента по категориям, тегам или по введенному тексту и возможность фильтрации контента по дате.

## 1.2 Обоснование необходимости автоматизации

Электронная схема — это сочетание отдельных электронных компонентов, таких как резисторы, конденсаторы, индуктивности, диоды, транзисторы и интегральные микросхемы, соединённых между собой.

Электронные схемы могут изображать разными способами, но наиболее распространены следующие:

1. Изображение электронной схемы на бумаге. Плюсы данного подхода – это доступность и простота. Минусы – можно неточно изобразить какой-либо компонент, бумага не надежный источник хранения данных и информацию в таков виде трудно быстро передать в другое место (как минимум нужно отсканировать или сфотографировать).
2. Изображение электронной схемы на компьютерной или мобильной программе. Плюсы данного подходы – это быстрота создания схемы так как основные компоненты уже встроены в программу и все что нужно это принести их на рабочую область. Минусы – данные программы нужно предварительно установить на устройство и большинство таких программ не поддерживают возможность экспорта электронной схемы в разных форматах.
3. Изображение электронной схемы на специализированном сайте. Данный подход лучше предыдущего, так как что бы пользоваться сайтом достаточно иметь две вещи: доступ в интернет и браузер. Что бы передать электронную схему, достаточно скопировать и отослать ссылку на нее. Единственный минус данного подхода — это то что нужен доступ к интернету.

Существует множество аналог:

* EasyEDA – сайт имеет много возможностей для изображения электронных схем, экспорта и конвертации в различные форматы. Данный сайт создан не для новичков, так как интерфейс будет понятен только опытным инженерам;
* Circuit Sims – сайт не имеет возможности экспорта электронных схем, только их изображения.
* EveryCircuit – хороший аналог с возможностью проверки электронных схем, но он не бесплатный в использовании.

Данный проект отличается от аналогов: интуитивно понятным интерфейсом, возможностью добавлять описания к схеме и просматривать другие схемы. Тем самым можно быстро найти нужную схему или похожую, и основываясь на ней сделать схему для себя – это уменьшит время на проектирования электронной схемы, тем самым с экономит время.

## 1.3 Определение данных и их представление

Форматы входных и выходных данных являются частью этапа проектирования. Входные форматы должны быть разработаны с учетом максимального удобства для пользователя и минимальной возможности ошибок. Порядок переменных и форматы данных, привычные для пользователя, помогут избежать ошибок и облегчат использование программ. В данном проекте входными данными являются: курсор мыши, настройки поиска, данные авторизации, электронные компоненты и текст.

Выходные спецификации могут сильно различаться. Иногда даются четкие инструкции и выходные данные подгоняются под определенный стандарт. Однако часто вообще отсутствуют какие-либо указания. Выходные данные подчас представляют собой: электронная схема и описанием.

# 2 Вычислительная система

## 2.1 Обоснование выбора языка программирования

В качестве языка программирования, выбранного для разрабатываемого проекта выбран язык программирования C#. Преимуществами данного языка являются:

1) C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности как FCL, так и CLR.

2) С# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами.

3) C# является мощным объектным языком с возможностями наследования и универсализации.

4) C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования: общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от С++ к C#.

5) Сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства С++ как указатели, адресация, разыменование, адресная арифметика.

6) Благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java. Эффективность кода даже повышается, поскольку исполнительная среда CLR представляет собой компилятор промежуточного языка, в то время как виртуальная Java-машина является интерпретатором байт-кода.

7) При помощи библиотеки EF Framework, осуществляется быстрый и удобный способ доступа к базе дынных.

## 2.2 Обоснование выбора среды разработки

Проект был реализован в среде быстрой разработки Visual Studio 2015.

Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня.

Преимущества Visual Studio 2015:

* использование вычислительных мощностей локального компьютера и облака;
* простая реализация общих задач и индивидуальный подход;
* быстрое создание высококачественного кода;
* функция поддержки нескольких мониторов.

## 2.3 Требования к конфигурации программного и аппаратного обеспечения

Данный проект разработан для ОС Windows 7. Достоинства:

1) Данную ОС устанавливают на большей массе современных ноутбуков, нетбуков и стационарных компьютеров, поэтому намного проще выполнить восстановление windows-системы после вирусов.

2) Высокая стабильность работы.

3) Интерфейс является крайне удобным. Интерфейс пользователя «семерки» в сравнении с Windows XP и Vista отличается яркостью красок и стилями оформления.

4) Windows 7 имеет высочайший уровень устойчивости к ошибочным и неосторожным действиям пользователей.

5) Перечень настроек Windows 7 весьма богат – начиная от графики и заканчивая правами доступа и настройкой безопасности.

6) Данная ОС совместима с современными распространенными программами.

Минимальные системные требования:

* ОС: Windows 7/8/10;
* процессор: Intel Core i3-3110M 2.40GHz;
* оперативная память: 2 ГБ (рекомендуется 4 ГБ).

# 3 ПРОЕКТИРОВАНИЕ

## 3.1 Проектирование интерфейса

Сайт должен состоять из 5 основных страниц:

1. Страница «Регистрации», содержит поля для ввода электронной почты и пароля, и кнопкой «зарегистрироваться».
2. Страница «Авторизации», содержит поля для ввода электронной почты и пароля, и кнопкой «войти», так же на странице есть 3 кнопки для входа на сайт через социальные сети (Вконтке, Tweeter, Facebook).
3. Главная страница, на которой расположены посты (название, автор, дата и время создания и часть описания). Слева располагаются категории схем, справа теги с текстовым полем для их поиска. Сверху страницы располагается текстовое поле для поиска постов.
4. Страница просмотра поста, в начале располагается электронная схема, над ней заголовок, автор, категория, теги и дата создания. Под электронной схемой описания и рейтинг. В самом низу располагаются комментарии.
5. Страница пользователя, слева-сверху располагается фото пользователя справа от него медальки пользователя, а снизу его посты.

## 3.2 Инфологическая модель предметной области

Электронные схемы сильно распространены инженерии микросхем. Гораздо удобнее нарисовать электронную схему в начале, а после сделать макет по схеме. Схемы можно хранить в базе данных – это обеспечит удобный поиск схем, что упростит последующее создания электронной схемы.

Проанализируем объекты реального мира. Для формирования концептуальной модели необходимо провести идентификацию объектов сущности базы данных.

Необходимо спроектировать базу данных электронных схем-сайт, информация которой будет использоваться для хранения электронных схем пользователей.

При проектировании БД необходимо учитывать следующее:

* к одному посту пользователь может добавить только одну схему;
* к одному посту можно добавить неограниченное количество тегов и комментариев;
* пользователь может просмотреть все свои посты и посты других пользователей;
* предусмотреть механизм ролей («Admin», «User»);
* у каждого поста должна быть категория из фиксированного списка;
* каждый пост имеет положительную и отрицательную отметку;
* должна иметься авторизация и регистрация.

Таким образом база данных содержит следующие сущности как Пользователи, Посты, Теги, Категории, Комментарии, Медальки.

Связь между сущностями ЯХТЫ и МОДИФИКАЦИИ, связывает яхту с ее модификациями. Данная связь позволяет связывать с одной ЯХТОЙ более одной МОДИФИКАЦИИ, причем каждая МОДИФИКАЦИЯ должна быть связана с какой-либо ЯХТОЙ. Также данная связь показывает, что каждая МОДИФИКАЦИЯ может принадлежать только одной ЯХТЕ.

Сущность ПРОДАЖИ имеет связь с такими сущностями как ЯХТЫ, ПОЛЬЗОВАТЕЛИ. Связь между сущностями ПРОДАЖИ и ПОЛЬЗОВАТЕЛИ, связывает пользователя и покупаемую им яхту**.** Данная связь предназначена для получения пользователем чека. Связь между сущностями ПРОДАЖИ и ЯХТЫ, связывает номер яхты, которую покупает пользователь. Данная связь предназначена для сохранения данных о покупаемой пользователем яхте.