**ООО «CMS Future»**

**УТВЕРЖДАЮ УТВЕРЖДАЮ**

Генеральный директор Генеральный директор

ООО «CMS Future» ООО «CCMS»

**\_\_\_\_\_\_\_\_**Григорьев А.В. **\_\_\_\_\_\_\_\_\_\_**Саевский О.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2021 г. «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

**«САПР для проектирования модели виртуального города»**

**ООО «CMS** **Future»**

**«Приложение CMS»**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

на 100 листах

Действует с «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2021 г.

**СОГЛАСОВАНО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

СОДЕРЖАНИЕ

**1 Общие сведения**

1.1 Полное наименование системы и его условное обозначение

1.2 Наименование предприятий разработчика и заказчика

1.3 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы

1.4 Плановые сроки начала и окончания работы по созданию системы

1.5 Сведения об источниках и порядке финансирования работ

1.6 Порядок оформления и предъявления заказчику результатов работ по созданию системы, по изготовлению и наладке отдельных средств и программно-технических комплексов системы

**2 Назначение и цели создания системы**

2.1 Назначение системы

2.2 Цели создания системы

**3 Характеристика объекта автоматизации**

**4 Требования к системе**

4.1 Требование к системе в целом

4.2 Требования к функциям, выполняемым системой

4.3 Требования к видам обеспечения

**5 Состав и содержание работ по созданию системы**

**6 Порядок контроля разработки и приемки**

**7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

**8 Требования к документированию**

**9 Источники разработки**

1 ОБЩИЕ СВЕДЕНИЯ

**1.1 Полное наименование системы и ее условное обозначение**

Полное наименование системы: приложение для проектирования системы управления содержимым.

Краткое наименование системы: приложение система управления содержимым.

**1.2 Наименование предприятий разработчика и заказчика**

Заказчик системы: ООО «CMS Future».

Адрес заказчика: г. Донецк, 11000, Ленинский проспект, 3.

Телефон: +3805553535.

Разработчик системы: ООО «CCMS».

Адрес разработчика: г. Донецк, 83000, Ленинский проспект, 138а.

Телефон: +38053535555.

**1.3 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы**

Основанием для разработки системы «Приложение для проектирования системы управления содержимым» являются приведенные ниже документы:

- Контракт №10/100055 от 01.09.2021 г. на выполнение подготовительных работ по созданию системы управления содержимым.

Документы были утверждены заказчиком с главой команды разработчиков у нотариуса вместе со всеми действиями по закону.

**1.4 Плановые сроки начала и окончания работ по созданию системы**

Плановый срок начала работ по созданию системы: 01.09.2021 г.

Плановый срок окончания работ по созданию системы: 01.09.2022 г.

**1.5 Сведения об источнике и порядке финансирования работ.**

Источник финансирования: Главный начальник ООО «CMS Future».

Порядок финансирования: ООО «CMS Future» изначально предоставляет ООО «CCMS» бюджет в $ 30 000, необходимый для проведения комплекса работ по созданию системы автоматизированного проектирования для CMS.

**1.6 Порядок оформления и предъявления заказчику результатов работ по созданию системы, по изготовлению и наладке отдельных средств и программно-технических комплексов системы**

«CMS» сдается в эксплуатацию заказчику в установленные настоящим техническим заданием сроки.

Акт сдачи-приема системы автоматизированного проектирования осуществляется уполномоченной комиссией в составе представителей заказчика и исполнителя.

Порядок предъявления системы, ее испытаний и окончательной приемки определен пунктом 6 настоящего ТЗ.

Совместно с предъявлением системы производится сдача разработанного исполнителем комплекта документации.

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы

**Что должна уметь система:**

1. Система должна обеспечивать работу с CMS, получать наглядное представление об объектах, удобно перемещаться и осматривать структуру проекта.
2. Система должна обеспечить автоматизацию разработки сайтов на всех уровнях проектирования.
3. Система должна осуществлять работу всей системы, а также всех её модулей, всех её драйверов, всех её устройств ввода-вывода и программ безопасности 24/7 без перебоев.
4. Система должна сократить срок проектирования сложных систем сайтов в четыре раза.
5. Система должна быть предназначена для драйверов, модулей, программ, оборудования и устройств, которыми она будет управлять и для которых будут подходить её параметры.
6. Система должна повысить качество проектируемого продукта и конструкторской документации.
7. Система должна обладать простотой и удобством.
8. Система должна позволить снизить требования к квалификации проектировщика.
9. Система должна иметь удобный, интуитивно понятный пользовательский интерфейс: крупный текст, структурированное горизонтальное иерархическое меню, панели инструментов, состояния, горячие клавиши.
10. Система должна осуществлять ввод-вывод и управление данными.
11. Система должна быть оснащена средствами визуализации процесса достижения цели.
12. Система должна осуществлять программную поддержку работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей и др.).
13. Система должна осуществлять фиксацию информации в БД.
14. Система должна планировать и организовывать процесс обработки запросов.
15. Система должна обеспечить быструю настройку содержимого веб приложения.
16. Система должна содержать обширную элементную базу, соответствующую новейшим разработкам систем управления содержимым.
17. Система должна иметь возможность расширения элементной базы.
18. Система должна предоставлять возможность получения данных в различных форматах.
19. Система должна осуществлять передачу информации между различными внутренними устройствами.
20. Система должна корректно отслеживать и записывать все данные, которые она собирает.
21. Система должна быть бесплатной.
22. Система должна обеспечить автоматизацию разработки сайтов на всех уровнях проектирования.
23. Использование системы должно позволять увеличить скорость разработки проекта, применить новейшие технические решения, повысить удобство для разработчиков и владельцев продукта.
24. Система должна позволить снизить требования к квалификации разработчика.
25. Система должна повысить качество проектируемого продукта и конструкторской документации.
26. Необходимо предоставить возможность быстрого развертывания сайта на сервере.
27. Необходимо предоставить разработчику выбор технологий среди некоторого перечня.
28. Система должна иметь понятный, легко читаемый исходный код.
29. Система должна уметь запускать сервер одной командой.
30. Пользователь должен сам выбирать порт сети.
31. Администратор должен иметь возможность смены пароля для входа в админ панель.
32. Система должна иметь поддержку плагинов.
33. Необходимо реализовать авторизацию по принципу OAuth 2.0.
34. Необходимо реализовать удобную работу в JQuery и Ajax.
35. Необходимо иметь уже установленный шаблон стилей Bootstrap.
36. Система должна обеспечить быструю настройку содержимого сайта.
37. Система должна содержать обширную элементную базу, соответствующую новейшим разработкам систем управления содержимым.
38. Система должна содержать все необходимые файлы и настройки для быстрого проектирования
39. Необходимо предоставить возможность комментирования и оценивание записей.
40. Необходимо предоставить видео уроки для работы с данной CMS.
41. Система должна предоставлять базовую админ панель, с реализацией CRUD системы.
42. Необходимо реализовать авторизацию для администратора сайта с целью защиты данных от взлома.
43. Система должна иметь поддержку плагинов.
44. Необходимо предоставить документацию разработчикам.
45. Система должна иметь свой API для взаимодействия.
46. Система должна поддерживать различные шаблоны, темы, плагины.
47. Система должна поддерживать различные форматы, такие как \*.png, \*.jpg, \*.pdf, \*.jpeg, \*.docx.
48. Система должна уметь работать с другими API.
49. Система должна уметь работать с Composer.
50. Система должна быть написана на PHP 8.0.
51. В системе должна быть представлена возможность смены версии языка PHP.
52. Система должна иметь ORM для работы с базой данных.
53. Система должна иметь MySQL СУБД.
54. Система должна иметь быстрый доступ к данным.
55. Системные требования для использования системы управления содержимым:

- процессор Intel Core i3 ГГц;

- Microsoft Linux/Windows 10;

- оперативная память 4 Гб;

- свободного места на диске 1000 Мб;

- видеопамять 256 Мб.

1. Для работы системы должен быть выход в интернет.
2. Система должна занимать мало места памяти жесткого диска.
3. Система должна использовать минимальное количество оперативной памяти.
4. Система должна быть легко переносимой между ОС.
5. Система должна иметь удобный, интуитивно понятный пользовательский интерфейс.
6. У системы должна быть главная страница.
7. У системы должен быть раздел с документацией и помощью.
8. Система должна иметь адаптивную верстку.
9. Система должна поддерживать различные шрифты, видео, изображения.
10. Система должна иметь набор инструкций по использованию.
11. Система должна помочь в разработке сайтов.
12. Система должна сэкономить время и цену на разработку сайтов.
13. В приложении должна быть реализована обработка исключительных ситуаций.
14. Система должна уведомлять пользователя об ошибках.
15. В системе должны быть методы для пользовательской обработки ошибок.
16. Система должна быть совместима с DOCKER контейнером.
17. Система должна запускаться быстрее 5 секунд.
18. Нельзя выбирать собственную ORM.
19. Нельзя удалять установленные пакеты Composer.
20. Версия PHP не может быть ниже 7.2.
21. База данных может использовать только СУБД MySQL.
22. Нельзя редактировать файлы системы.
23. Система не может работать с API, требующих SSH сертификат при запуске по локальной сети.
24. Система должна быть оптимизирована под смартфоны
25. Разные версии системы должны обладать совместимостью.
26. Система должна предоставлять полную поддержку работы с Excel файлами.
27. Система должна содержать интуитивно понятный и эффективный инструментарий.
28. Система должна содержать защиту от хакерских атак.
29. Система должна отображать данные динамически.
30. Система должна поддерживать фильтрацию данных.
31. Система должна позволять добавлять 3D модели и анимации.
32. Система должна иметь средства работы с геолокацией.
33. В системе должна быть реализована возможность обмена данными через облачные хранилищ.
34. Система должна содержать модули для создания реалистичного видео и симуляции прогулки по городу на основе данных проектной модели.
35. Система должна иметь удобные средства для просмотра смоделированной 3D-сцены.
36. Разработанные приложения на базе системы должны быть доступны в следующих версиях популярных браузеров:

- Mozilla Firefox версии 14 и выше;

- Google Chrome версии 20 и выше;

- Internet Explorer 11;

- Safari версии 5 и выше;

- Яндекс.Браузер 16 и выше.

1. В системе должна присутствовать обработка исключений с выводом соответствующих уведомлений.
2. Тестирование системы должно быть возможно и без подключения к сети.
3. Система должна иметь иметь обратную совместимой с предыдущими версиями.
4. В системе должно быть предусмотрена установка плавающей лицензии, которая не привязывается к конкретному рабочему месту, но в один момент времени только один пользователь может пользоваться ей, остальные получат отказ.
5. При первом открытии программного продукта система должна открывать окно знакомства с пошаговыми инструкциями по её настройке и предлагать ссылки на учебные пособия и видеоматериалы.
6. Система должна обеспечивать возможность автоматического создания комплекса спецификаций, расчётов количества объектов моделей, а также анализ метаданных.
7. Система должна иметь unit тестирование по умолчанию.
8. Система должна поддерживать работу авторизированных пользователей, предлагая ввод логина и пароля при входе в систему.
9. Система должна работать на базе движка Nguinx.

**Описание моделей деятельности успешных компаний отрасли**

CMS (англ. Content Management System) — это система управления контентом сайта. На профессиональном жаргоне CMS ещё называют «движок сайта».  На CMS работает больше половины сайтов в интернете. Примеры CMS: WordPress, Joomla, OpenCart.

CMS осуществляет контроль за версиями содержимого, что очень удобно при коллективной работе над контентом. В его роли могут выступать текстовые документы, видео, аудиофайлы и базы данных.

WCMS (веб-версии систем управления) отвечают за показ страниц сайта, мгновенно компилируя их содержимое из двух частей: непосредственно самого контента и шаблонов оформления, которые хранятся в базе данных ресурса. Саму систему CMS можно условно разделить на две составляющие:

База данных, в которой хранится контент.

Элементы визуализации (шаблоны).

В зависимости от характера работы шаблона, CMS дифференцируют на три типа:

Автономные. Система самостоятельно обрабатывает данные. Такие CMS подходят для статических сайтов, структура которых состоит из набора неизменяемых блоков.

Интерактивные. Применяются для управления динамичных сайтов.

Гибридные. Сочетают в себе функции динамичных и автономных систем.

Благодаря этой системе, сам сайт невозможно представить в виде "книги" с уже напечатанными страницами. Ее попросту не существует, а процессы верстки и издания начинаются в тот момент, когда пользователь протягивает руку к пустой книжной полке, то есть отправляет запрос. Сайт — это набор компонентов, которые собираются на ходу в одно целое при поступлении соответствующего запроса. За "сборку" страниц и отвечает CMS. Чем быстрее программа генерирует ответ на запрос, тем выше оценивается ее работа. Естественно, существуют простенькие сайты, в которых можно обойтись и без CMS, но организовать работу вручную для больших ресурсов попросту невозможно.

CMS можно назвать незаменимым помощником для обладателя сайта, который не имеет специальных навыков для работы с ним. Система выступает в роли посредника, который "переводит" команды человека и предоставляет ему лаконичный инструментарий для управления ресурсом.

Основные этапы моделирования содержимого:

1. Проектирование внешнего вида содержимого. Создание базового шаблона.

2. Подготовка материала. Создание СУБД, используемой по умолчанию.

3. Моделирование объектов БД. Происходит по специально разработанной технологии.

4. Моделирование маршрутизации. Создание маршрута главной страницы и страницы администратора с помощью шаблона по умолчанию.

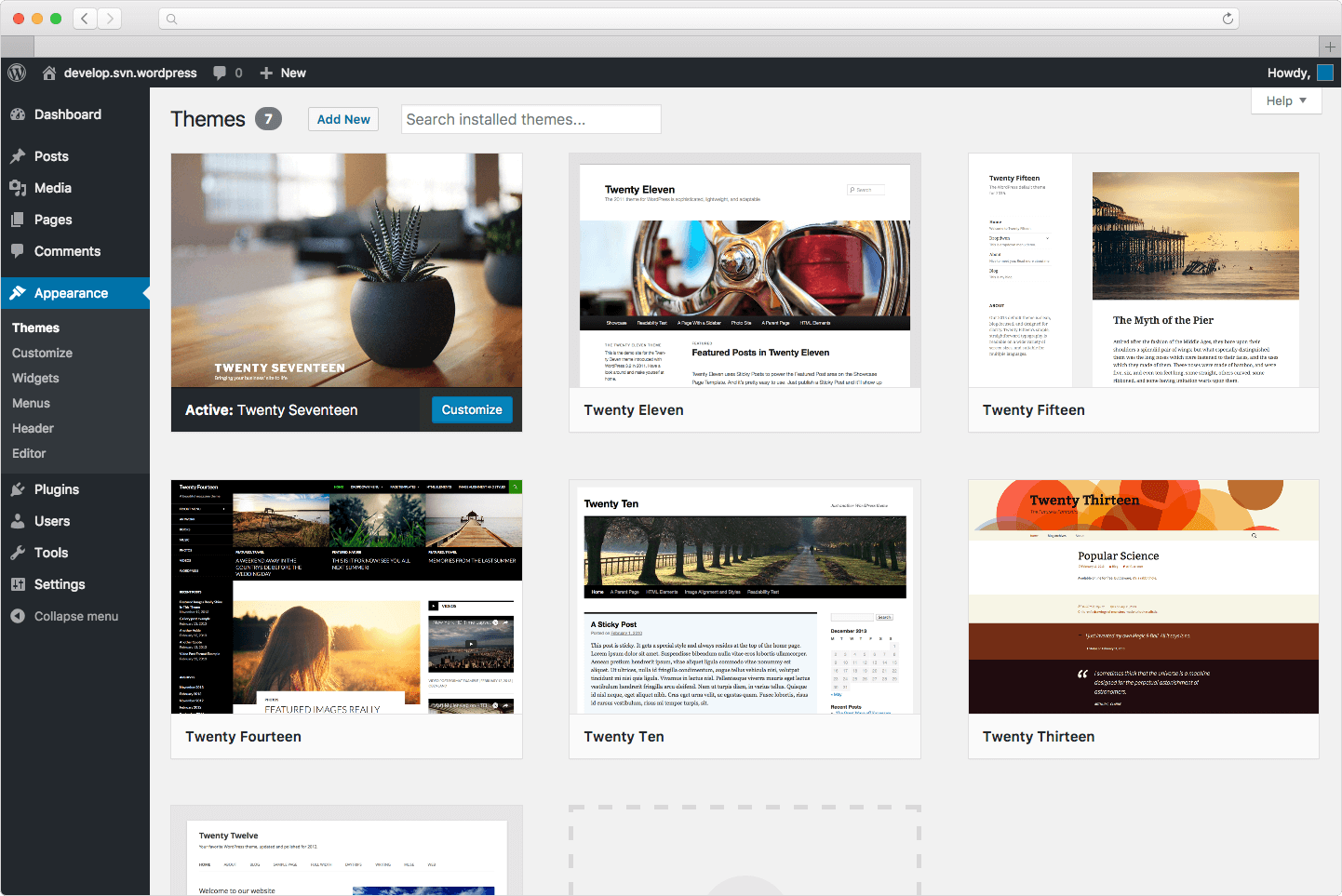
5. Развертывание содержимого на сервере.

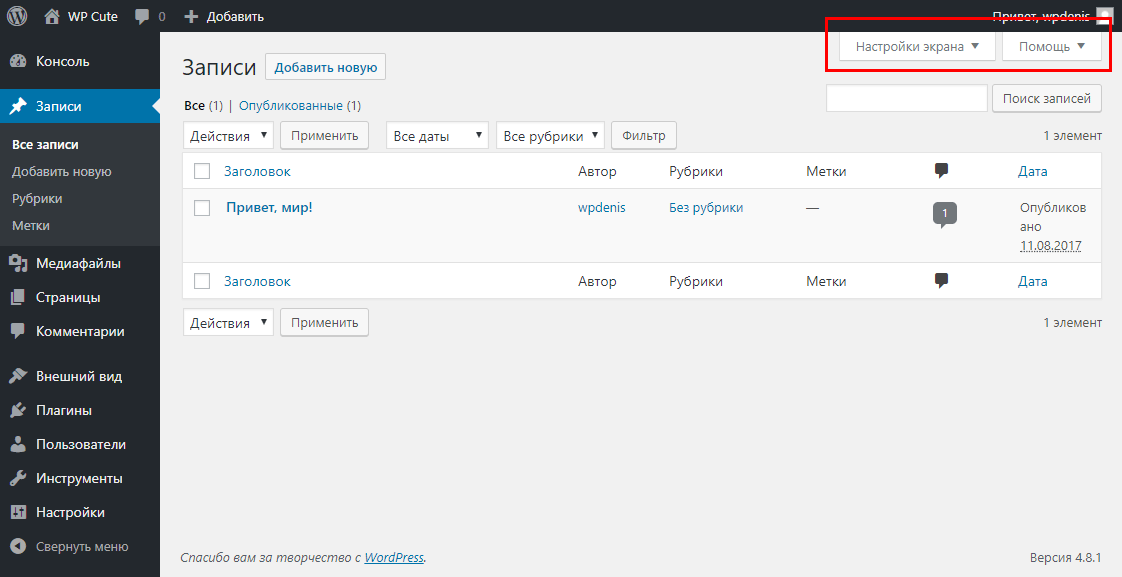
6. Изменение содержимого, добавление новых шаблонов.

7. Тестирование. Производится визуальная оценка содержимого и измеряется её вид при необходимости.

**Описания моделей деятельности успешных компаний отрасли**

**WordPress** — самая распространённая CMS в мире. Она имеет свободную лицензию и развивается благодаря огромному пользовательскому сообществу. По умолчанию WordPress предоставляет лишь минимальные возможности для организации блога, но благодаря расширеням пользователи превращают простой движок в универсальный инструмент для создания различных сайтов, от лендингов до многостраничных порталов с миллионной аудиторией.





Преимущества:

WordPress разворачивается в один клик почти на всех хостингах, поэтому с установкой системы проблем точно не будет. Вы можете инсталлировать её и вручную, скачав дистрибутив с сайта разработчика — инструкций и руководств по этому вопросу достаточно. Учебные материалы понадобятся и для выполнения первоначальной настройки движка, которая включает в себя изменение общих параметров работы и установку нескольких обязательных плагинов.

Гибкость.  Гибкость WP обеспечивают плагины, доступные в соответствующем разделе административной панели и в каталогах на других сайтах. Расширения решают целый ряд важных задач:

обеспечивают безопасность — например, защищают от спама и несанкционированного доступа.

помогают оптимизировать сайт под требования поисковиков;

добавляют новые возможности для взаимодействия пользователей с интерфейсом.

Наличие визуального редактора Gutenberg. Он сделал систему ещё более удобной для использования. После создания и предварительной настройки Gutenberg становится основным инструментом кастомизации внешнего вида и управления контентом. Он предлагает выстраивать страницы из функциональных блоков, которые выполняют разные задачи:

добавление текстового контента: заголовков, абзацев, цитат, списков;

вставку медиа-файлов: отдельных картинок, галереи, аудиозаписей и видеороликов;

интеграцию элементов разметки;

размещение виджетов и приложений, созданных сторонними разработчиками.

Недостатки:

WordPress не идеален. Созданные на нём сайты часто подвергаются взлому, но это не проблемы движка, а следствие его распространённости: им пользуются миллионы веб-мастеров, среди которых многие не озадачиваются глубокой проработкой вопросов безопасности.

Ещё один потенциальный недостаток — снижение производительности из-за использования большого количества плагинов. С этим сталкиваются все начинающие веб-мастера. WordPress предлагает миллион возможностей, всё хочется попробовать, но в какой-то момент дополнения начинают слишком грузить сервер. Проблема решается переносом сайта на более мощную и дорогую инфраструктуру или отказом от плагинов — в конце концов, открытый исходный код WordPress позволяет реализовывать на нём отдельные функции без установки дополнений. Правда, для этого потребуются навыки программирования или бюджет для привлечения специалистов.

Иногда проблемы создают обновления системы. Ни в коем случае нельзя ставить их в автоматическом режиме — только вручную, после создания резервной копии. Обновление движка может привести к тому, что бесплатные плагины перестанут работать из-за несоответствия версий. С платными решениями такие сложности обычно не возникают — разве что иногда приходится недолго ждать их апдейтов.

**Joomla**. Joomla является весьма интересным решением для быстрого создания интернет-магазина или сайтов другой направленности. Данная CMS считается одной из самых востребованных и популярных. Создать сайт, используя этот движок, можно достаточно быстро, без дополнительных проблем. Joomla позволяет создавать довольно сложные проекты людям, не обладающим огромным опытом работы с PHP.



Преимущества:

1. Доступность

CMS Joomla является бесплатной, что делает ее востребованным вариантом для начинающих предпринимателей, которым пока сложно выделить деньги на покупку дорогостоящих профессиональных движков. Сегодня можно увидеть большое количество условно бесплатных CMS, в данном случае у вас не будет возможности создать уникальный сайт, внося корректировки в процессе его создания. Такие действия будут расценены, как нарушения изначального соглашения. А Joomla позволяет модифицировать сайты по собственному усмотрению, любые ограничения в данном случае отсутствуют.

2. Простота установки

Joomla можно достаточно быстро и легко установить, процесс создания сайта является удобным и понятным. Если вам нужные новые шаблоны - в интернете представлено огромное количество различных решений, есть даже масштабные, платные шаблоны. Для того, чтобы оперативно разобраться с особенностями функционирования Joomla, достаточно просмотреть несколько видеоуроков, которыми переполнен Интернет.

3. Система является открытой

Никаких закрытых компонентов, CMS Joomla отличается полностью открытым исходным кодом. Пользователи могу редактировать его, учитывая собственные предпочтения - это одно из основных достоинств Joomla.

У вас будет возможность не только редактировать исходный код Joomla, вносить необходимые коррективы в дизайн, а создавать сайты на заказ на Joomla, чтобы получить определенное вознаграждение. Сегодня сложно найти альтернативную CMS с таким набором характеристик.

Есть только одно ограничение для тех, кто собирается использовать в работе Joomla. Нельзя вносить изменения в лицензию, нарушать авторские права.

4. Огромное количество компонентов и модулей

В середине 2005 года появилась первая версия данной CMS. Прошло достаточно много времени, Joomla активно развивалась, были созданы инновационные компоненты и модули. Появились разнообразные галереи, интеграция с новыми системами для проведения платежей.

Такой подход позволяет использовать Joomla для создания невероятно стильных сайтов, если при работе возникла проблема с определенными компонентами, можно просто найти несколько групп фрилансеров, где размещены любительские разработки членов огромного сообщества Joomla.

5. Сайт, созданный на Joomla, является достаточно простым и практичным.

Сегодня можно найти большое количество CMS, используемых для создания сайтов. Практически все они отличаются достаточно сложной "админкой", что делает управление ресурсом весьма затрудненным.

Пользователи таких систем сталкиваются с проблемами при выполнении элементарных действий. Необходимо 5-7 отдельных действий, чтобы создать простую страницу - это достаточно сложно и неудобно.

Joomla отличается простой интуитивной "админкой", весь необходимый функционал доступен всем пользователям. Достаточно всего 2 кликов, чтобы выполнить требуемое действие.

6. Русификация различных модулей и расширений

Энтузиасты продолжают заниматься переводом популярных модулей Joomla на русский язык, это дает возможность существенно упростить работу с ними в "админке". Если вы нашли модуль, в котором изначально нет возможности выбора русского языка - просто поищите в Google, вы обязательно найдете русификатор.

7. Система регулярно обновляется, появляются новые полезные модули

Joomla постоянно обновляется, появляются новые модули, разработчики защищают систему от покушений хакеров, "латая" разнообразные дыры. Учитывая популярность CMS, ее обновление происходит достаточно часто.

8. Помощь в решении сложных вопросов

Если при работе Joomla вы столкнулись с определенной проблемой, можно спокойно найти ответ на свой вопрос - есть огромное количество групп поддержки пользователей данной CMS. Всегда найдется информация, которая будет важна и полезна именно вам.

9. Хостинг

Joomla является универсальной CMS и для нее подходит большинство современных хостингов.

Недостатки:

Структура некоторых элементов системы не доработана;

Система очень плохо защищена от постороннего вмешательства;

Периодически страдает индексация сайта;

Иногда сложно обновить систему, сразу после появления очередного обновления;

Отсутствует официальная служба поддержки.

**Drupal** — бесплатный движок, который можно скачать с сайта разработчика или установить через панель администратора на хостинге. Второй способ предпочтительнее, так как при его использовании нет необходимости создавать базу данных и выполнять предварительную настройку системы — достаточно прикрепить к аккаунту на хостинге зарегистрированное доменное имя и выбрать его для развёртывания системы управления контентом.

Чтобы привыкнуть к интерфейсу Drupal, нужно время. Он достаточно логичен, но обилие функций и настроек усложняет задачу для начинающих веб-мастеров.

При установке Drupal уже имеет стандартный комплект модулей, с помощью которых можно управлять сайтом. В их числе:

* Tracker для отслеживания новых товаров.
* Block для вывода контента на страницах.
* Color для изменений цветовой схемы сайта, если такую возможность предоставляет тема оформления.
* Image для обработки изображения.
* Node для добавления новых материалов.
* Search для организации поиска по сайту.
* System для управления системными параметрами.
* User для управления зарегистрированными пользователями.
* Locale для выбора языковых пакетов.
* Path для переименования ссылок.
* Update Manager для проверки обновлений.



Преимущества:

* Открытый исходный код и структура, обеспечивающая максимальную гибкость системы при настройке.
* Удобная установка в один клик из личного кабинета га хостинге.
* Большой выбор тем оформления и дополнений для расширения функциональности.
* Простая панель управления, открытая для кастомизации.
* Встроенная система кэширования, обеспечивающая ускорение загрузки страниц.
* Развитое пользовательское сообщество с русскоязычными площадками, где разработчики делятся опытом.
* Мультиязычность.

Недостатки:

Работа с модулями — вообще одна из главных проблем при использовании Drupal. Чтобы реализовать какую-либо функцию, обычно приходится устанавливать два или три дополнения. Это связано с тем, что модули на Drupal не дублируют, а используют возможности друг друга. С расширениями связана и другая сложность — нестабильная работа после обновлений движка. Не все модули получают апдейты сразу, поэтому часто приходится немного ждать, прежде чем обновлять CMS.

Drupal также крайне требователен к ресурсам. Для обеспечения высокой производительности нужен мощный хостинг, в противном случае никакое кэширование страниц не спасёт от медленной загрузки контента.

**Идентификация совладельцев**

К интересантам со стороны Исполнителя относятся: менеджер проекта, эксперты в области написания ОС и дальнейшей её поддержке, эксперты в области написания технического документа (инструкции по эксплуатации), специалисты в области моделирования и проектирования моделей зданий.

К интересантам со стороны Заказчика относятся: пользователи, специалисты в области установки оборудования, инвесторы.

2.2 Цели создания системы

Система создаётся для обеспечения автоматизации разработки виртуального города, предназначенного для геоинформационных систем и геопорталов.

Система создается для повышения эффективности моделирования 3D-макета города, внедрения новейших технологий в данных процесс, облегчения труда разработчиков виртуального города и т.д.

Система создается для обеспечения производительности, надежности и безопасности исполнения проекта виртуального города.

Система создается для получения экономического выигрыша при использовании системы, путем увеличения производительности труда программистов и эффективности работы алгоритмов моделирования, применяемых в программе.

Система должна сократить срок проектирования трёхмерной модели города на предприятии в четыре раза.

Система должна повысить качество проектируемого продукта и конструкторской документации.

Система должна позволить снизить требования к квалификации проектировщика.

Акторы

Для бизнес требований:

- Система;

- Разработчик;

- Локальный сервер;

- Сайт;

- PHP;

- Docker;

Требований пользователя:

- Сервер

- Разработчик

- Composer

- Bootstrap

- JavaScript

- Ajax

Функциональных требований поведения:

- Система

- API

- Composer

- ORM

- MySQL

|  |  |
| --- | --- |
| ***Таблица для бизнес требований*** |  |
| ***Актор*** | ***Действие*** |
| ***Разработчик*** | ***Устанавливает систему*** |
| ***Система*** | ***Устанавливает необходимые зависимости*** |
| ***Разработчик*** | ***Запускает систему*** |
| ***Система*** | ***Запускает локальный сервер и развертывает сайт*** |
| ***Сайт*** | ***Подгружает Html код и запрашивает интерпретацию PHP кода*** |
| ***Локальный сервер*** | ***Запускается и читает Html код*** |
| ***PHP*** | ***Интерпретирует свой код в HTML разметку*** |
| ***Разработчик*** | ***Упаковывает проект в Docker*** |
| ***Docker*** | ***Собирает все таблицы MySQL, все зависимости Composer и код разработчика в контейнер*** |
| ***Разработчик*** | ***Публикует Docker контейнер на сайте*** |
| ***Docker*** | ***Развертывается на сайте, устанавливает все зависимости*** |
| ***Сайт*** | ***Развертывает приложение*** |

|  |  |
| --- | --- |
| ***Таблица для требований пользователя*** |  |
| ***Актор*** | ***Действие*** |
| ***Разработчик*** | ***Выбирает зависимости для своего сайта*** |
| ***Composer*** | ***Устанавливает зависимости*** |
| ***Разработчик*** | ***Использует Ajax код*** |
| ***Ajax*** | ***Использует свой функционал для динамической работы с сайтом*** |
| ***Разработчик*** | ***Пишет предустановленные CSS стили в Bootstrap*** |
| ***Bootstrap*** | ***Берет готовые шаблоны из сети*** |

|  |  |
| --- | --- |
| ***Таблица для функциональных требований*** |  |
| ***Актор*** | ***Действие*** |
| ***Разработчик*** | ***Использует API метод системы*** |
| ***Система*** | ***Получает запрос и дает ответ*** |
| ***Разработчик*** | ***Пишет запрос к ORM на добавление данных*** |
| ***ORM*** | ***Обращается к MySQL сервера с данными*** |
| ***MySQL*** | ***Добавляет данные в таблицу или выдает ответ об ошибке*** |
| ***Разработчик*** | ***Пишет обработку ошибки используя API системы*** |
| ***Система*** | ***Получает запрос и дает ответ*** |

5. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

Система автоматизированного проектирования (САПР) – автоматизированная система, реализующая информационную технологию выполнения функций проектирования, представляет собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из персонала и комплекса технических, программных и других средств автоматизации его деятельности.

Работы по созданию системы выполняются в три этапа:

1) Проектирование. Разработка эскизного проекта. Разработка технического проекта (продолжительность — 5 месяцев);

2) Разработка рабочей документации. Адаптация программ (продолжительность — 5 месяцев);

3) Ввод в действие (продолжительность — 1-2 месяца).

Конкретные сроки выполнения стадий и этапов разработки и создания САПР определяются планом выполнения работ, являющимся неотъемлемой частью договора на выполнение работ по настоящему техническому заданию.

Перечень организаций - исполнителей работ, определение ответственных за проведение этих работ организаций определяются договором.

По окончанию всех работ по созданию системы автоматизированного проектирования «САПР виртуального города» составляется акт сдачи-приемки. Заказчику предоставляется полный комплект технической документации.

Экспертиза технической документации должна производиться комиссией, состоящей из уполномоченных представителей заказчика и исполнителя.

Система должна разрабатываться в виде последовательности самостоятельно функционирующих версий, каждая из которых должна поставляться заказчику для проведения испытаний. Заказчик (либо его уполномоченный представитель) должен составить рецензию на поставленную версию, содержащую коррективы и изменения, и предоставить ее команде разработчиков.

6 ПОРЯДОК КОНТРОЛЯ РАЗРАБОТКИ И ПРИЕМКИ

Система подвергается испытаниям следующих видов:

1. Предварительные испытания;

2. Опытная эксплуатация;

3. Приемочные испытания.

Предварительные испытания:

* Предварительные испытания системы проводятся для определения ее работоспособности и решения вопроса о возможности приемки системы в опытную эксплуатацию.
* Предварительные испытания проводятся приемочной Комиссией Заказчика, назначаемой из уполномоченных на то представителей Заказчика и Исполнителя, в соответствии с Программой и методикой испытаний.
* Результаты испытаний фиксируются в протоколе.

Состав, объем и методы предварительных испытаний системы определяются документом «Программа и методика испытаний», разрабатываемым на стадии «Рабочая документация».

Опытная эксплуатация:

* Опытная эксплуатация проводится для определения правильности функционирования системы и возможности ее передачи на приемочные испытания.
* Результаты опытной эксплуатации фиксируются в Акте о завершении опытной эксплуатации и допуске системы к приемочным испытаниям.

Состав, объем и методы опытной эксплуатации системы определяются документом «Программа опытной эксплуатации», разрабатываемым на стадии «Ввод в действие».

Приемочные испытания:

* Приемочные испытания системы проводятся с целью определения ее готовности к вводу в постоянную (промышленную) эксплуатацию.
* Приемочные испытания проводятся приемочной Комиссией Заказчика, назначаемой из уполномоченных на то представителей Заказчика и Исполнителя. Комиссия в своей работе руководствуется положениями контракта, настоящего Технического задания, Технического проекта, а также Программы и методики испытаний, разрабатываемой Исполнителем и утверждаемой Заказчиком.
* Приемочные испытания проводятся путем выполнения контрольных заданий, перечисляемых в Программе и методике испытаний.
* По результатам выполнения контрольных заданий Комиссией составляется Акт проведения приемочных испытаний, являющийся основанием для передачи системы в постоянную (промышленную) эксплуатацию.

Состав, объем и методы приемочных испытаний системы определяются документом «Программа и методика испытаний», разрабатываемым на стадии «Ввод в действие» с учетом результатов проведения предварительных испытаний и опытной эксплуатации.

Требования к приемке работ по стадиям приведены в таблице.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Стадия испытаний | Участники испытаний | Место и срок проведения | Порядок согласования документации | Статус приемочной комиссии |
| Формирование требований к ОС. | Команда разработчиков | ООО «New Horizons» |  |  |
| Разработка концепции ОС. | Команда разработчиков | ООО «New Horizons» |  |  |
| Техническое задание. | Команда разработчиков | ООО «New Horizons» |  |  |
| Эскизный проект. | Команда разработчиков | ООО «New Horizons» |  |  |
| Технический проект. | Команда разработчиков | ООО «New Horizons» |  |  |
| Рабочая документация. | Команда разработчиков | ООО «New Horizons» |  |  |
| Ввод в действие. | Команда разработчиков | ООО «New Horizons» |  |  |
| Сопровождение |  | ООО «New Horizons» |  |  |

Каждая из поставленных заказчику версий системы должна быть испытана заказчиком (либо его уполномоченными представителями) и рецензирована.

По окончанию всех работ по созданию операционной системы «ОС управления выращивания грибов в теплице» составляется акт сдачи-приемки. Заказчику предоставляется полный комплект технической документации.

Заказчик имеет право тестировать систему в течение 100 дней с момента составления акта.

После тестирования заказчик должен принять работу, либо в письменном виде изложить причину отказа. В случае обоснованного отказа исполнитель обязуется доработать систему. По истечению указанного срока претензии со стороны заказчика не принимаются.

Статус приемной комиссии определяется заказчиком.

7 ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ СИСТЕМЫ В ДЕЙСТВИЕ

Выделяют следующие основные стадии создания и этапы разработки системы автоматизированного проектирования:

1. Формирование требований к САПР;
2. Разработка концепции САПР;
3. Техническое задание;
4. Эскизный проект;
5. Технический проект;
6. Рабочая документация;
7. Ввод в действие;
8. Сопровождение САПР.

Как правило, этапы стадий формирования требований к системе автоматизированного проектирования объединяют с этапами разработки технического задания, а этапы разработки концепции - с этапами эскизного проектирования. Также к приведенным стадиям создания САПР добавляют стадию подготовки к началу проекта.

Стадия I. Организация проекта

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Результат** | **Дней** |
| Заключение контракта | Контракт на разработку системы заключен | 10 |
| Согласование процедур управления | Процедуры управления проектом и устав проекта согласованы | 5 |
| Сбор команды проекта | Команда проекта сформирована | 5 |
| Обучение членов проектной команды | Компетенция участников проекта соответствует требованиям. Этап реализуется по необходимости | 10 |

Стадия II. Формирование требований и разработка технического задания

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Результат** | **Дней** |
| Системно-аналитическое обследование объекта автоматизации | Проведены интервью с функциональными и IT-специалистами Заказчика.  Собрана информация о:  требуемых отчетах, показателя, измерениях  возможных системах-источниках данных  системном ландшафте  сетевом окружении | 20 |
| Анализ и обработка полученной информации | Сформированы:  альбом показателей  альбом измерений и иерархий  альбом отчетности  определены источники данных для каждого показателя | 20 |
| Разработка концептуальной модели данных | Концептуальная модель данных | 15 |
| Разработка технического задания | Техническое задание и приложения к нему | 20 |
| Согласование и утверждение | Согласованное и утвержденное техническое задание | 10 |

Стадия III. Эскизный проект

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Результат** | **Дней** |
| Определение общей функциональной и технической архитектур | Общее описание функциональной и технической архитектур | 10 |
| Разработка логической модели данных | Логическая модель данных.  Определен состав сущностей области постоянного хранения (System of Record, Summary Area, Data Marts).  Окончательно сформирован состав реализуемых показателей, измерений и отчетов. | 30 |
| Обоснование выбора программного обеспечения и технической инфраструктуры | ПО выбрано. Утвержден план формирования необходимой технической инфраструктуры | 10 |
| Разработка предварительного регламента взаимодействия информационных систем | Предварительные регламенты взаимодействия согласованы с разработчиками и администраторами систем-источников данных.  Определен состав сущностей области временного хранения (Staging Area) и области обмена данными (Data Exchange Interface). | 20 |
| Эскизное проектирование процессов ETL | Общее описание процессов извлечения данных, алгоритмов трансформации, загрузки и агрегации данных | 10 |
| Эскизное проектирование интерфейсов пользователя | Общее описание интерфейсов ввода и предоставления данных | 5 |
| Оформление эскизного проекта | Пояснительная записка к эскизному проекту | 15 |
| Согласование и утверждение | Согласованная и утвержденная пояснительная записка к эскизному проекту | 5 |

Стадия IV. Технический проект

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Результат** | **Дней** |
| Определение функциональной и технической архитектур | Описание функциональной архитектуры (включая описание каждой функции, задачи, методов реализации).  Описание технической архитектуры (включая описание размещения технических средств, условий эксплуатации, режима функционирования, организации резервного копирования и т.д.) | 15 |
| Формирование плана развертывание системного ландшафта | План развертывания системного ландшафта | 10 |
| Разработка физической модели данных | Физическая модель данных - описание таблиц БД, индексов, секций и других объектов БД | 10 |
| Разработка и согласование регламентов взаимодействия информационных систем | Согласованный и утвержденный набор регламентов взаимодействия, включающих описание интерфейсов, периодичности и т.п. | 10 |
| Проектирование процессов ETL | Описание процессов извлечения данных, алгоритмов трансформации и обеспечения качества данных, процессов загрузки и агрегации данных | 20 |
| Проектирование интерфейсов пользователя | Описание бизнес-слоя данных, интерфейсов ввода и предоставления данных, разграничения прав доступа | 15 |
| Оформление технического проекта | Пояснительная записка к техническому проекту | 20 |
| Согласование и утверждение | Согласованная и утвержденная пояснительная записка к техническому проекту | 5 |

Стадия V. Рабочая документация

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Результат** | **Дней** |
| Разработка рабочей документации на систему и на её части | Разработаны следующие документы:  Ведомость эксплуатационных документов  Ведомость машинных носителей информации  Паспорт  Общее описание системы  Технологическая инструкция  Руководство пользователя  Описание технологического процесса обработки данных (включая телеобработку)  Инструкция по формированию и ведению базы данных (набора данных)  Состав выходных данных (сообщений)  Каталог базы данных  Программа и методика испытаний (ПИМ)  Спецификация  Описание программ  Текст программ | 40 |
| Разработка или адаптация программ | Развернуты экземпляры БД. Созданы необходимые объекты БД  Разработаны процессы ETL и процессы обеспечения качества данных. Выставлено расписание запуска процессов  Реализованы дополнительные приложения  Реализованы витрины данных и отчетность  Настроены профили пользователей и прав доступа | 60 |
| Согласование и утверждение | Согласованная и утвержденная рабочая документация | 15 |

Стадия VI. Ввод в действие

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Результат** | **Дней** |
| Подготовка объекта автоматизации к вводу системы в действие | Создание у Заказчика службы сопровождения системы (при необходимости) | 10 |
| Подготовка персонала | Обучение пользователей и администраторов системы | 10 |
| Комплектация системы поставляемыми изделиями | Закупка и завоз необходимого оборудования и программного обеспечения. Для снятия рисков поставки данный этап обычно выполняется на предыдущей стадии | 20 - 100 |
| Строительно-монтажные работы | Оборудование смонтировано в выделенном для этого помещении и подключено к каналам передачи данных | 20 |
| Пусконаладочные работы | Прошла наладка технических и программных средств. ПО системы перенесено в зону тестирования/промышленной эксплуатации. Настроена система резервного копирования.  Проведена загрузка исторических данных в систему. Запущены процессы извлечения данных из систем-источников | 20 |
| Проведение предварительных испытаний | Испытания системы на работоспособность и соответствие техническому заданию в соответствии с ПИМ проведены.  Устранение неисправностей и внесение изменений в документацию в соответствии с протоколом испытаний проведено.  Акт приёмки системы в опытную эксплуатацию оформлен и подписан | 10 |
| Проведение опытной эксплуатации | Опытная эксплуатация проведена. Устранение неисправностей, доработка ПО и дополнительная наладка технических средств проведены | 30 - 60 |
| Проведение приёмочных испытаний | Испытания системы в соответствии с ПИМ проведены.  Устранение неисправностей и внесение изменений в документацию проведено.  Акт передачи системы в промышленную эксплуатацию оформлен и подписан | 5 |
| Завершение работ | Акт завершения работ подписан | 5 |

Стадия VII. Сопровождение

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Результат** | **Дней** |
| Выполнение работ в соответствии с гарантийными обязательствами | Выявленные недостатки системы устранены | 275 |
| Послегарантийное обслуживание | Система работает стабильно, без сбоев. Выявленные недостатки устранены | - |

После проведения экспертизы сторонами подписывается акт сдачи-приемки, после чего система вводится в эксплуатацию.

Заказчик и исполнитель должны активно сотрудничать. Исполнитель должен предоставлять отчет о проделанной работе заказчику в виде очередной версии системы.

8 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

Для разрабатываемой системы должны быть выпущены комплекты документов, соответствующих требованиям ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы».

Также необходимыми являются документы «Описание программы», «Руководство программиста», «Руководство оператора».

Содержание документов является общим для всех видов автоматизированных систем и при необходимости может дополняться Разработчиком в зависимости от особенностей конкретно создаваемой Системы. Допускается включать в документы дополнительные разделы и сведения, объединять и исключать разделы.

В составе технорабочего проекта разрабатывается документация по общесистемным решениям, организационному, техническому, информационному и программному обеспечению, а также проектно-сметная документация. В состав эксплуатационной документации входит документация по информационному, программному, техническому и метрологическому обеспечению, а также проектно-сметная документация. В соответствии с ГОСТ 34.201-89, п. 1.3.1, табл. 2, виды документов, разрабатываемых на стадиях Технического проекта и Рабочей документации и имеющих отношение к проектносметным, выполняются Проектной организацией.

Вся рабочая документация, разработанная применительно к данному конкретному проекту, должна быть на русском языке.

Количество экземпляров проектной и эксплуатационной документации, предоставляемой Заказчику, определяется Договорами с Поставщиком оборудования и Разработчиком проекта, однако в любом случае должно быть не менее четырех.

1. ИСТОЧНИКИ РАЗРАБОТКИ
2. Коростылёв Р.И., Еремин И.Е. Электронная карта с использованием реалистичных 3D-моделей зданий // Ученые заметки ТОГУ. – 2013. – Т. 4, № 3. – с. 67 – 71.
3. 1. Гречищев, А. Трехмерное моделирование и фотореалистичная визуализация городских территорий / А. Гречищев, В. Бараниченко, С. Монастырев, А. Шпильман // ArcReview. – 2003. – №2 – С.12-13.
4. Дубинин М.В., Мишаченко К.Г., Еремин И.Е. Реалистичная модель городского пространства // Ученые заметки ТОГУ. – 2014. – Т. 5, № 4. – с. 1379-1384.
5. Основные стратегии создания 3D-моделей городов [Электронный ресурс]/ Александр Бондарец. – 2010. – Режим доступа: <http://gis-lab.info/qa/3dcities.html>.
6. Ulm K., 2003 Reality-based 3D city models with CyberCity-Modeler (CCmodeler) and laserscanner data. VI Conference on Optical 3D Measurement Techniques – Gruen/Kahmen (Eds), 2003. – p.32-39.

ПРИЛОЖЕНИЕ А

**Глоссарий**

1. **Трёхмерная графика** — раздел компьютерной графики, посвящённый методам создания изображений или видео путём моделирования объёмных объектов в трёхмерном пространстве.
2. **Трёхмерное моделирование** — создание трёхмерной математической модели сцены и объектов в ней.
3. **3D-сцена** - это пространственное отображение всех 3D элементов и трехмерных тел в 3D виде окна текущего чертежа.
4. **3D-модель** – это виртуальная объемная геометрическая модель объекта (геометрическая фигура), являющая собой набор поверхностей, размещенных в трехмерном координатном пространстве.
5. **Трёхмерное пространство** — геометрическая модель материального мира, в котором мы находимся. Это пространство называется трёхмерным, так как оно имеет три однородных измерения – длину, ширину и высоту, то есть трёхмерное пространство описывается тремя единичными ортогональными векторами.
6. **Текстурирование** – проецирование растровых или процедурных текстур на поверхности трёхмерного объекта в соответствии с картой UV-координат, где каждой вершине объекта ставится в соответствие определённая координата на двухмерном пространстве текстуры.
7. **UV-преобразование**, или развёртка в трёхмерной графике, — соответствие между координатами на поверхности трёхмерного объекта (X, Y, Z) и координатами на текстуре (U, V).
8. **Цифровая модель рельефа (ЦРМ)** — средство цифрового представления трехмерных пространственных объектов (поверхностей или рельефов) в виде трехмерных данных, образующих множество высотных отметок (отметок глубин) и иных значений аппликат (координаты Z) в узлах регулярной или нерегулярной сети или совокупность записей горизонталей (изогипс, изобат) или иных изолиний.
9. **Наземный лазерный сканер (НЛС)** — это съёмочная система, измеряющая с высокой скоростью (от нескольких тысяч до миллиона точек в секунду) расстояния от сканера до поверхности объекта и регистрирующая соответствующие направления (вертикальные и горизонтальные углы) с последующим формированием трёхмерного изображения (скана) в виде облака точек.
10. **Облако точек** — набор вершин в трёхмерной системе координат.
11. **Аэрофотосъёмка** — фотографирование территории с определённой высоты от поверхности Земли при помощи аэрофотоаппарата, установленного на атмосферном летательном аппарате (самолёте, вертолёте, дирижабле и пр. или их беспилотном аналоге)] с целью получения, изучения и представления объективных пространственных данных на участках произведенной съемки.
12. **Пространственные данные** (географические данные, геоданные) — данные о пространственных объектах и их наборах.
13. **Архитектурный стиль** — совокупность характерных черт и признаков архитектуры.
14. **Инфраструктура** – комплекс отраслей хозяйства, обслуживающих промышленное или другое производство, а также население.

ПРИЛОЖЕНИЕ Б

UML-диаграммы



Рисунок 1 – Диаграмма классов



Рисунок 2 – Диаграмма вариантов использования основного функционала система

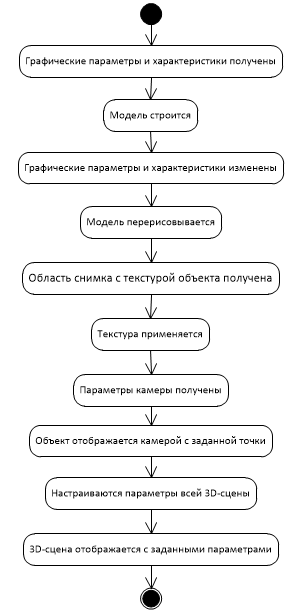


Рисунок 3 – Диаграмма состояний моделирования объекта городской среды

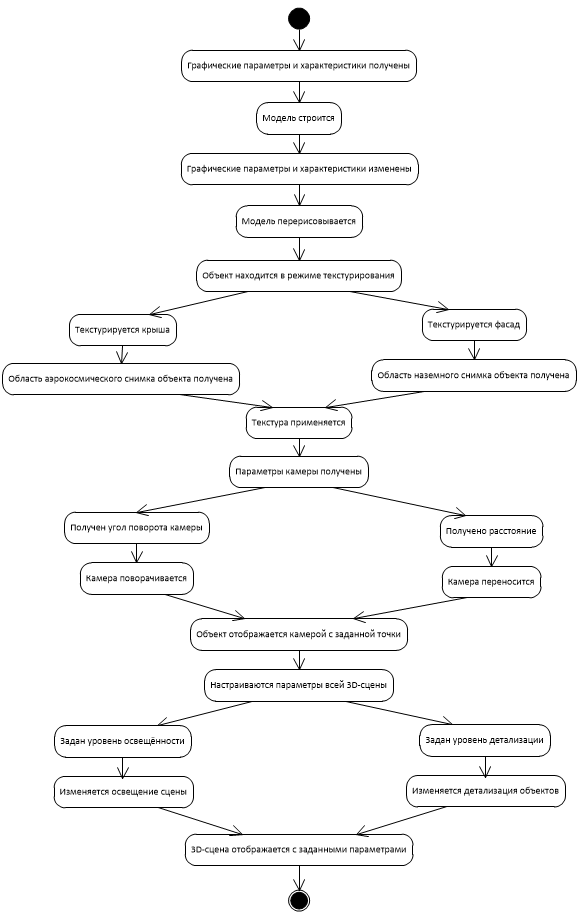
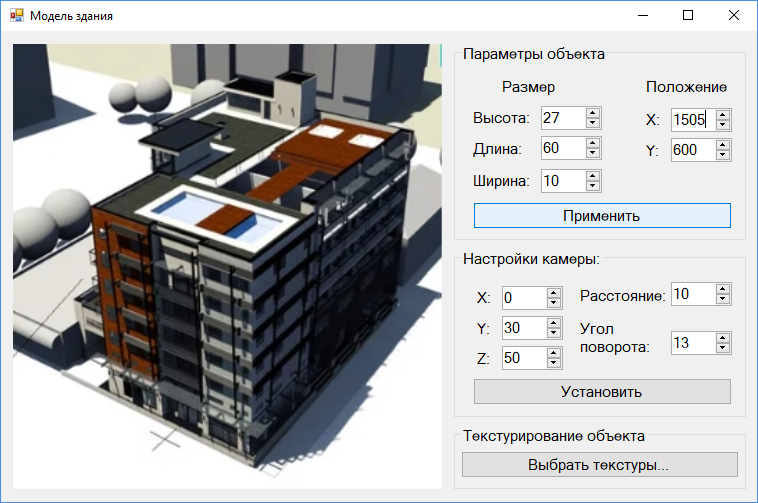


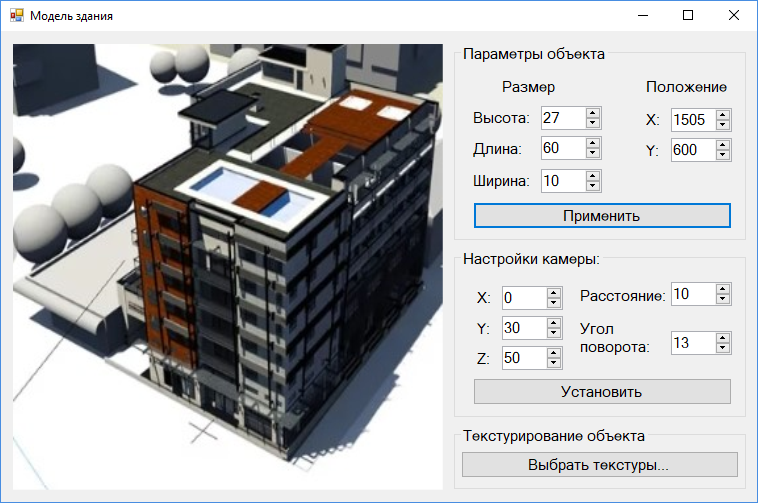
Рисунок 4 – Диаграмма действий моделирования объекта городской среды

ПРИЛОЖЕНИЕ В

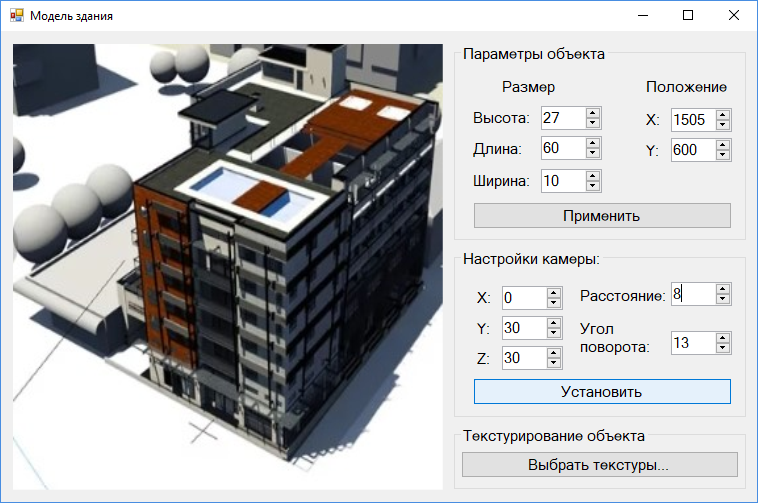
Экранные формы

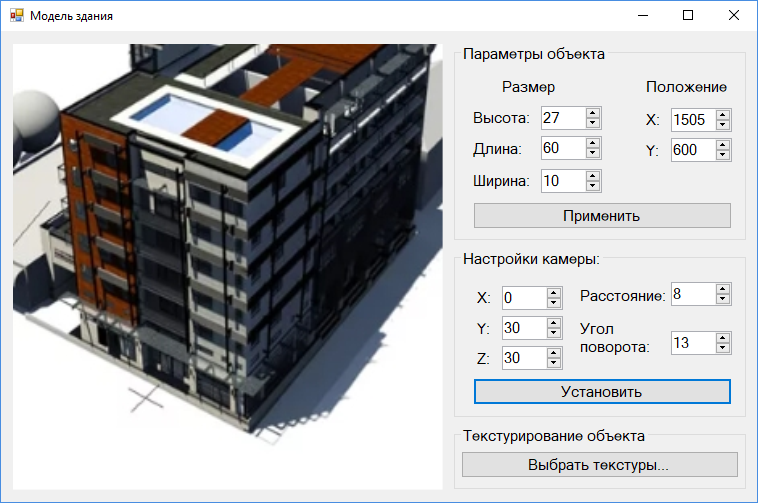
1. **Горизонтальный прототип (вариант 1)**
2. Установка параметров объекта: длины, высоты и ширины здания, а также положения его на карте.



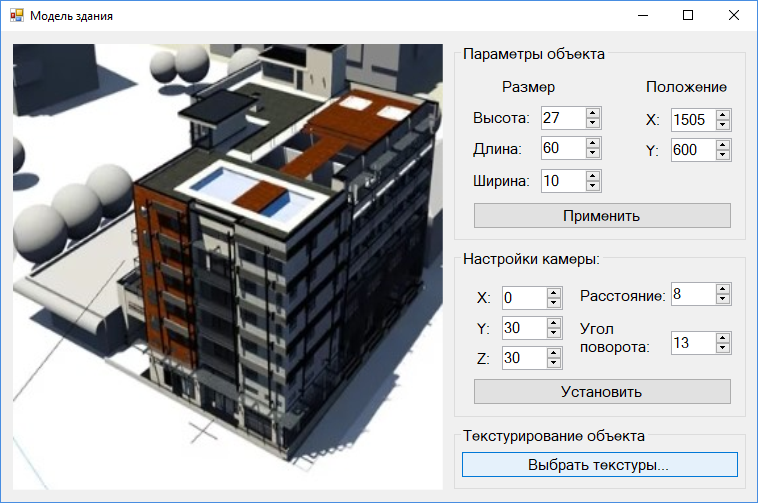


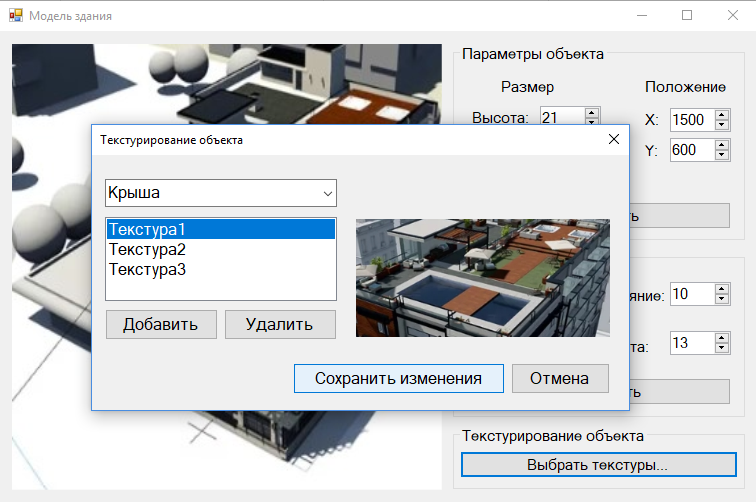
1. Установка настроек камеры: точки камеры, расстояния до объекта, угла поворота.





1. Выбор текстур для объекта.



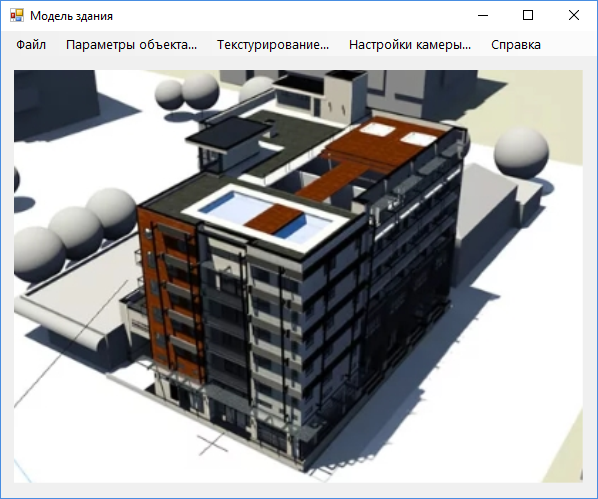


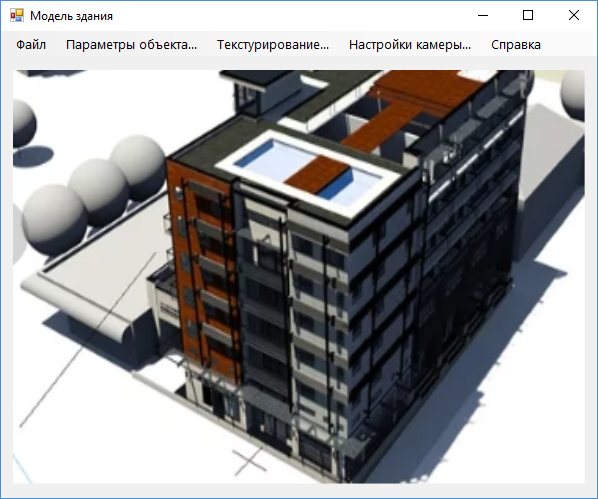
1. **Таблица достоинств и недостатков**

|  |  |
| --- | --- |
| **Достоинства** | **Недостатки** |
| Интерфейс интуитивно понятен и прост | Диагональный баланс не соблюдается |
| Интерфейс обладает свойством естественности и гибкости | Избыточность элементов на главной форме |
| Быстрый доступ ко всем функциям | Отсутствие основного меню |
| Использование шрифтов крупных размеров без засечек | Отсутствие «горячих» клавиш |
| Использование подсветки и затенения интерактивных элементов | Не соблюдается требование группировки элементов |
| Легкость чтения и логическая согласованность | Наличие кнопок выбора представления окна в диалоговом окне |
| Группирование элементов по назначению | Отсутствие справки |
| Используется выравнивание | Отсутствие возможности сохранения модели |

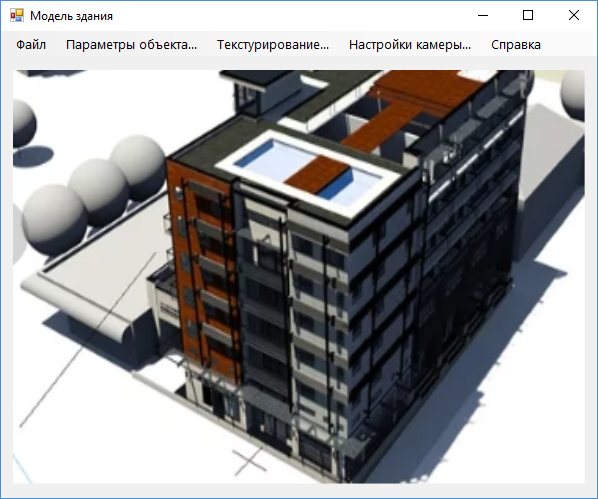
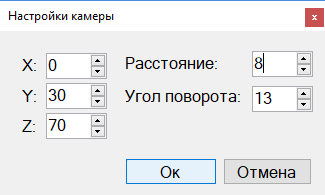
**3. Горизонтальный прототип (2 вариант)**

1. Установка параметров объекта: длины, высоты и ширины здания, а также положения его на карте.



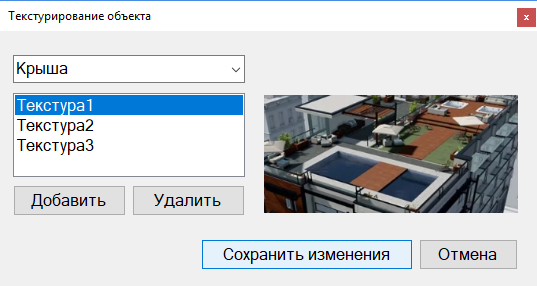


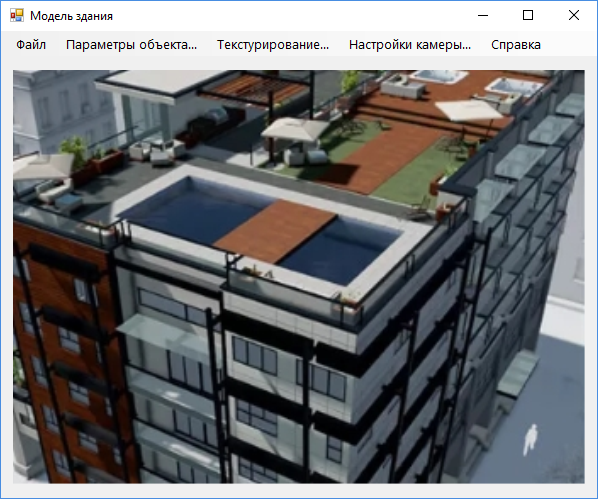
1. Установка настроек камеры: точки камеры, расстояния до объекта, угла поворота.



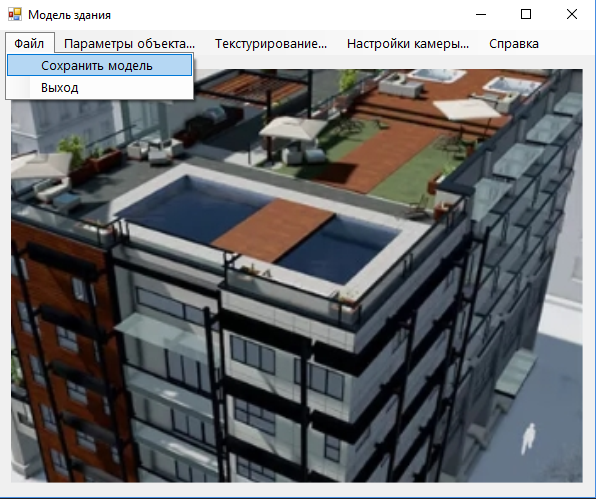


1. Выбор текстур для объекта.





1. Сохранение модели в программе.



**4. Таблица достоинств и недостатков**

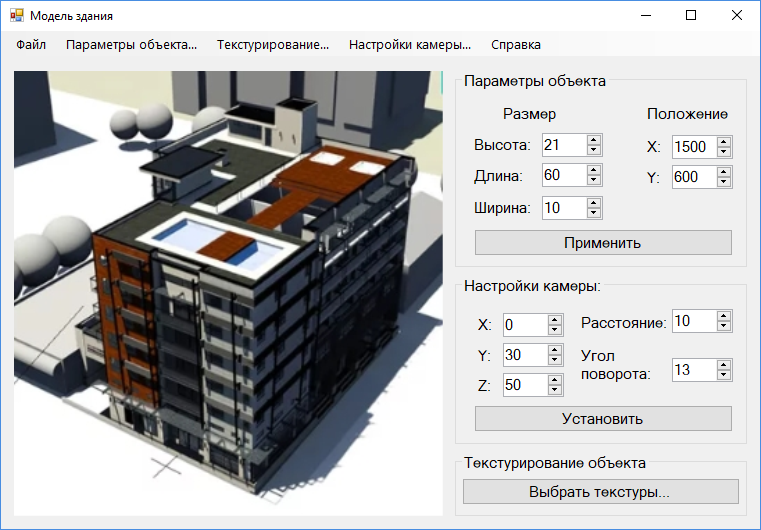
|  |  |
| --- | --- |
| **Достоинства** | **Недостатки** |
| Интерфейс интуитивно понятен и прост | Отсутствует быстрый доступ к функциям системы |
| Интерфейс обладает свойством естественности и гибкости | Отсутствие «горячих» клавиш |
| Отсутствует избыточность элементов на главной форме |  |
| Использование шрифтов крупного размера без засечек |  |
| Использование подсветки и затенения интерактивных элементов |  |
| Легкость чтения и логическая согласованность |  |
| Используется выравнивание |  |
| Используется основное меню |  |
| Присутствует пункт меню «Справка» |  |
| Имеется возможность сохранения модели |  |
| Используется принцип группировки |  |
| Соблюдается диагональный баланс |  |

**5. Описание комбинированного сценария**

Учитывая достоинства и недостатки описанных выше сценариев, которые имеют большое число общих достоинств, можно составить комбинированный сценарий, сочетающий в себе достоинства и исключающий недостатки предложенных ранее сценариев.

В результирующий прототип следует включить как основное меню, так и панель, обеспечивающую быстрый доступ к функциям системы, немного откорректировав её (обеспечить выполнения свойства группировки и диагонального баланса, сократить заголовки панели). Следует также назначить каждому пункту меню соответствующую ему «горячую» клавишу. Необходимо добавить возможность зума и поворота камеры при помощи прямого манипулирования.

Ниже приведен результирующий прототип главной формы приложения.



**5. Вертикальный прототип**

Реализация системы будет производится высокоуровневом языке С#, поддерживающем многие возможности низкого уровня, с применением объектно-ориентированной парадигмы программирования. Предполагается использование Visual Studio 2017 в качестве среды разработки, имеющей мощный инструментарий (набор компонентов), предоставляющий широкий спектр возможностей для разработки.

В качестве СУБД будет использована система PostgreSQL, которая поддерживает язык SQL и предлагает обширный набор возможностей для работы с базой данной.

Разработка интерфейса системы будет осуществляться с помощью технологии Win API, поддерживающей «подстройку» интерфейса под нужды конкретного пользователя.

Система должна корректно функционировать на IBM – совместимых PC со следующей системной конфигурацией:

• процессор Intel Core i5 ГГц;

• оперативная память 2 Гб;

• свободного места на диске 9 Гб;

• видеопамять 256 Мб.

на базе операционных систем семейства Win 32 и Win 64 (XP/Vista/7/8.1/10).