**Введение**

Тема разрабатываемого программного продукта является сайт для услуг хим.чистки или стирки различных вещей и одежды.

Созданный программный продукт рассчитана на аудиторию, которая хочет постирать или вычистить вещи хим. чисткой. Данный программный продукт представляет собой сайт для заказа услуг стирки или хим. чистки вещей или одежды. Сайт предоставляет возможность зарегистрироваться для входа в личный кабинет и осуществления в последующем заказа услуг хим.чистки или стирки.

Приведём краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нём можно ознакомиться с постановкой задачи. В подразделе «Инструменты обработки» рассмотрена среда, в которой был создан программный продукт. Здесь также установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачи.

В разделе «Проектирование задачи» рассмотрены основные аспекты разработки программного продукта. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе описан пользовательский интерфейс, составление алгоритмов процесса обработки информации, описана разработка системы справочной информации.

«Реализация задачи» − это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые использованы при реализации данного приложения. В этом разделе описаны функции пользователя и их структура.

«Заключение» содержит краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В разделе «Список используемых источников» приведён список используемых при разработке источников.

**1 Анализ задачи**

* 1. **Постановка задачи**

**1.1.1 Организационно-экономическая сущность задачи**

**1.1.2 Функциональные требования**

Сайт должен обеспечивать возможность выполнения следующих

функций гостя:

1) Просмотр страниц;

2) Просмотр цен;

3) Просмотр услуг;

4) Просмотр контактов;

5) Просмотр новостей и акций;

6) Просмотр способа оплаты;

7) Доступ к социальной сети;

Администратор (права администратора представлены всему персоналу

организации):

1) Добавление новой информации (новости, стоимость услуг)

2) Удаление информации (удаление ненужной или устаревшей

информации);

3) Редактирование новой информации (в случае опечатки или для того, чтобы дополнить какую-то информацию);

**1.1.3 Эксплуатационные требования**

Будет использоваться конструктор WordPress/

Для реализации динамических страниц должен использоваться язык PHP.

Требования к надежности: система может быть недоступна не более чем 24 часа в год. У администратора сайта должна быть возможность выгрузить и загрузить копию сайта.

Требования к интерфейсу: при разработке сайта должны быть использованы преимущественного белые и зеленые оттенки. Основные разделы сайта должны быть доступны с первой страницы. Грамотный пользовательский интерфейс. Сайт должен адаптироваться под компьютер, телефон и планшет.

* 1. **Диаграмма вариантов использования**

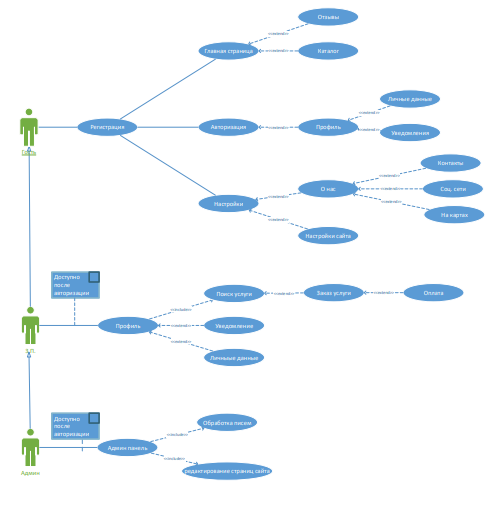


Рисунок 1 – Диаграмма вариантов использования

* 1. **Модель данных**

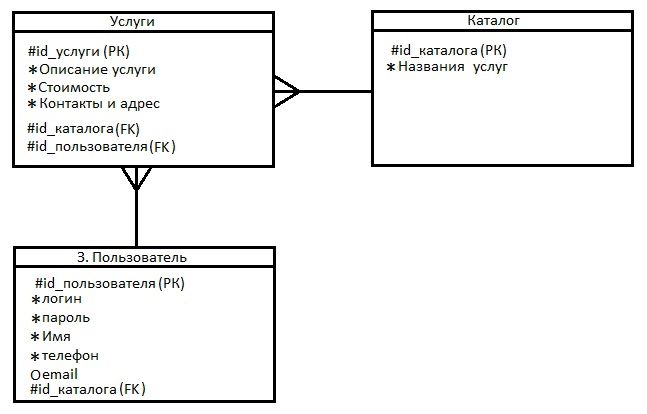


Рисунок 2 – Модель данных

**1.4 Техническое задание**

**1.4.1 Введение**

**1.4.1.1 Наименование программного продукта**

Наименование – «Химчистка «Чистый Рубеж»

**1.4.1.2 Ответственные лица**

Разработчик: Копачёв А.Р.

**1.4.1.3 Сроки сдачи проекта**

**1.4.1.4 Краткая характеристика области применения**

Сайт предназначен для людей, которым нужно очистить свои вещи.

**1.4.2 Основания для разработки**

**1.4.2.1 Основания для проведения разработки**

Основание для разработки является задание на тему

«Разработка интернет-сайта для организации, оказывающей услуги химчистки». Тема согласована с преподавателем, именуемым в дальнейшем заказчиком.

**1.4.2.2 Наименование и условное обозначение темы разработки**

Наименование темы разработки - «услуги химчистки».

**1.4.3 Требования к программе или программному изделию**

**1.4.3.1 Требования к функциональным характеристикам**

Сайт должен обеспечивать возможность выполнения следующих

функций гостя:

1) Просмотр страниц;

2) Просмотр цен;

3) Просмотр услуг;

4) Просмотр контактов;

5) Просмотр новостей и акций;

6) Просмотр способа оплаты;

7) Доступ к социальной сети;

Администратор (права администратора представлены всему персоналу

организации):

1) Добавление новой информации (новости, стоимость услуг)

2) Удаление информации (удаление ненужной или устаревшей

информации);

3) Редактирование новой информации (в случае опечатки или для

того, чтобы дополнить какую-то информацию);

**1.4.4 Исходные данные**

**1.4.5 Требование к надежности**

Система может быть недоступна не более чем 24 часа в год.

**1.4.5.1 Требование к обеспечению надежного (устойчивого)**

функционирования программы

Надежное (устойчивое) функционирование программы должно быть

обеспечено выполнением Заказчиком совокупности организационно-

технических мероприятий, перечень которых приведен ниже:

1. Организацией бесперебойного питания технических средств;

2. Использование лицензионного программного обеспечения;

3. Регулярным выполнением рекомендаций Министерства труда и

социального развития;

4. Регулярным выполнением требований ГОСТ 51188-98. Защита

информации. Испытания программных средств на наличие

компьютерных вирусов.

Требования к обеспечению надежного (устойчивого)

функционирования программы не предъявляются.

**1.4.5.2 Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30 минут при условии соблюдения условий технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

**1.4.5.3 Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

**1.4.6 Условия эксплуатации**

Любое устройство с выходом в интернет

**1.4.6.1 Требование к видам обслуживания**

Веб-сайт требует проведение каких-либо видов обслуживания.

Администратор будет обновлять, редактировать и удалять какую-то

информацию в приложении.

**1.4.6.2 Требование к численности и квалификации персонала**

Минимальное количество персонала, требуемого для работы сайта,

должно составлять не менее 1 штатной единицы – системный администратор.

**1.4.6.3 Требование к составу и параметрам технических средств**

Доступ к данному приложению может осуществляться с помощью любого

компьютера и мобильного устройства. Для хорошей работы сайт нужен

устойчивое Интернет-соединение.

**1.4.7.1 Требования к представлению сайта**

Для организации эффективной работы пользователя нужно создать

целостный программный продукт данной предметной области, в котором все

компоненты будут сгруппированы по функциональному назначению. При

этом необходимо обеспечить удобный графический интерфейс пользователя.

**1.4.7.2 Требования к структуре сайта**

Все названия разделов сайта, приведенные ниже, являются условными и могут корректироваться по соглашению с Заказчиком в ходе проектирования. При помощи системы управления сайтом структура и состав разделов сайта в дальнейшем могут быть изменены и дополнены.

**1.4.8 Требование к программной документации**

**1.4.8.1 Предварительный состав программной документации**

Состав программной документации должен включать в себя:

1. Постановка задачи;
2. Модель данных;
3. Диаграмма вариантов использования;
4. Диаграмма последовательности;
5. Диаграмма компонентов;
6. Техническое задание;

**1.4.9 Технико-экономические показатели**

Ориентировочная экономическая эффективность не рассчитываются.

Предполагаемое число использований программы в год – 365 сеансов

работы на одном рабочем месте.

**1.4.10 Стадии и этапы разработк**

**1.4.10.1 Стадии разработки**

Разработка должна быть проведена в три стадии:

1. Разработка технического задания;
2. Рабочее проектирование;
3. Внедрение;

**1.4.10.2 Этапы разработки**

На стадии разработки технического задания должен быть выполнен

этап разработки, согласования и утверждения настоящего технического

задания.

На стадии рабочего проектирования должны быть выполнены

перечисленные ниже этапы работ:

1. Разработка программы;
2. Разработка программной документации;

Испытание программы;

**1.4.11 Требования к надежности**

Система может быть недоступна не более чем 24 часа в год. Резервирование данных осуществляет хостинг-провайдер. У администраторасайта должна быть возможность выгрузить и загрузить копию сайта.Программа поставляется в виде программного продукта– на хостинге. Хостингпозволяет создавать и хранить контент на сайте.

**1.4.12 Требования к обеспечению надежного (устойчивого) функционирования программы**

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно- технических мероприятий, перечень которых приведен ниже:

1 Организацией бесперебойного питания технических средств;

2 Использование лицензионного программного обеспечения;

3 Регулярным выполнением рекомендаций Министерства труда и

социального развития;

4 Регулярным выполнением требований ГОСТ 51188-98. Защита

информации. Испытания программных средств на наличие

компьютерных вирусов.

**1.4.13 Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30 минут при условии соблюдения условий технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

**1.4.14 Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

**1.4.15 Условия эксплуатации**

Веб-сайт требует проведение каких-либо видов обслуживания. Администратор будет обновлять, редактировать и удалять какую-либо информацию на сайте.

**1.4.16 Требования к численности и квалификации персонала**

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц – системный администратор и конечный пользователь программы – оператор. Системный администратор должен иметь высшее профильное образование и сертификаты компании.

**2 Проектирования задачи**

**2.1 Выбор стратегии разработки и модели жизненного цикла**

Для выбора подходящей к условиям конкретного проекта модели ЖЦ ПС и систем Институтом качества программного обеспечения SQI рекомендуется использовать специальную процедуру. Данная процедура базируется на применении четырех таблиц вопросов.

Каждая из табл. 1 – 4 представляет одну из категорий классификации проектов. Каждый из вопросов (строка в таблице) предназначен для классификации анализируемого проекта по определенному критерию категории. Столбцы данных таблиц соответствуют обобщенным моделям ЖЦ, фактически представляющим стратегии разработки ПС. При этом под RAD-моделью подразумевается независимая RAD-модель, не встроенная в другие модели жизненного цикла.

Рассматриваемая процедура состоит из следующей последовательности шагов:

1 проанализировать отличительные черты проекта по критериям категорий, представленным в виде вопросов;

2ответить на вопросы по анализируемому проекту, отметив слова «да» или «нет» в соответствующих строках табл. 1 – 4. Если слов «да» или «нет» в строке несколько, необходимо отметить все из них (все «да» или все «нет»).

В качестве примера в табл. 1 выделены варианты ответов для проекта разработки сложного и критичного программного средства, требования к которому заранее не известны и будут уточняться по ходу разработки;

3 расположить по степени важности категории (таблицы) и/или критерии, относящиеся к каждой категории (вопросы внутри таблиц), относительно проекта, для которого выбирается модель ЖЦ;

4выбрать из моделей (см. табл. 1– 4) ту модель, которая соответствует столбцу с наибольшим количеством отмеченных ответов с учетом их степени важности (с наибольшим количеством отмеченных ответов в верхней части приоритетных таблиц). Выбранная модель ЖЦ является наиболее приемлемой для анализируемого проекта.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Характеристика | Жизненный цикл | Недостатки и достоинства | Для каких проектов |
| 1.Каскадная | Однократный проход этапов  разработки. Данная стратегия основана на полном определении всех требований к разрабатываемому программному средству или системе в начале процесса разработки. Каждый этап разработки начинается после завершения предыдущего этапа. Возврат к уже выполненным этапам не предусматривается. |  | **Достоинства** 1) стабильность требований в течение ЖЦ разработки;  2)необходимость только одного прохода этапов разработки, что обеспечивает простоту применения стратегии;  3) простота планирования, контроля и управления проектом;  4) доступность для понимания заказчиками.  **Недостатки** 1) сложность полного формулирования требований в начале процесса  разработки и невозможность их динамического изменения на протяжении ЖЦ  2) разрабатываемые ПС или системы обычно слишком велики и сложны, чтобы все работы по их созданию выполнять однократно в результате возврат к предыдущим шагам для решения возникающих проблем приводит к увеличению финансовых затрат и нарушению графика работ  3) непригодность промежуточных продуктов для использования  4) недостаточное участие пользователя в процессе разработки ПС | 1) при разработке проектов с четкими, неизменяемыми в течение ЖЦ  требованиями и понятной реализацией;  2) при разработке проектов невысокой сложности  3) при выполнении больших проектов в качестве составной части моде-  лей ЖЦ |
| 2.Спиральная | Данная модель жизненного цикла характерна при разработке новаторских (нетиповых) систем. В начале работы над проектом у заказчика и разработчика нет четкого видения итогового продукта (требования не могут быть четко определены) или стопроцентной уверенности в успешной реализации проекта (риски очень велики). В связи с этим принимается решение разработки системы по частям с возможностью изменения требований или отказа от ее дальнейшего развития. Как видно из рис.3.4, развитие проекта может быть завершено не только после стадии внедрения, но и после стадии анализа риска |  | **Достоинства** 1)позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований; 2) обеспечивает большую гибкость в управлении проектом; 3) уменьшаются риски заказчика. Заказчик может с минимальными для себя финансовыми потерями завершить развитие неперспективного проекта.  **Недостатки** 1) увеличивается неопределенность у разработчика в перспективах развития проекта. Этот недостаток вытекает из предыдущего достоинства модели;  2) затруднены операции временного и ресурсного планирования всего проекта в целом. Для решения этой проблемы необходимо ввести временные ограничения на каждую из стадий жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа выполнена. План составляется на основе статистических данных, полученных в предыдущих проектах и личного опыта разработчиков. | 1) при разработке новаторских (нетиповых) систем. 2) 3) |
| 3.V-образная модель. | Модель представляет собой разновидность каскадной модели.  Данная модель поддерживает каскадную стратегию однократного выполнения  этапов процесса разработки ПС или систем и базируется на предварительном  полном формировании требований. В классической V-образной модели каждый шаг начинается после завершения предыдущего шага.  Отличием V образной модели от каскадной является то, что в ней выделены связи между шагами, предшествующими программированию, и соответствующими видами тестирования и испытаний. |  | **Достоинства** 1) планирование тестирования и испытаний на ранних стадиях разработ-  ки системы и программного средства;  2) упрощение аттестации и верификации промежуточных результатов  разработки;  3) упрощение управления и контроля хода процесса разработки. **Недостатки** 1) поздние сроки тестирования требований в жизненном цикле, что оказывает существенное влияние на график выполнения проекта при необходимости изменения требований;  2) отсутствие, как и в остальных каскадных моделях, действий, направленных на анализ рисков. | 1)Если требуется тщательное тестирование продукта, то V-модель оправдает заложенную в себя идею: validation and verification. 2)Для малых и средних проектов, где требования четко определены и фиксированы. 3)В условиях доступности инженеров необходимой квалификации, особенно тестировщиков. |
| 4.RAD-модель | представляет собой инкрементную модель, в которой множество разработок маленьких - маленьких кусков выбираются и развиваются одновременно для достижения большей картины. Кроме того, обрабатывается инкрементная модель, в которой основные характеристики, подлежащие разработке, делятся на более мелкие, выполнимые куски. |  | **Достоинства 1)** в модели используются современные методы моделирования данных; 2) + достоинства структурной эволюционной модели быстрого прототипирования. 3) требуется меньшее количество специалистов, т.к. специалисты хорошо владеют предметной областью 4) благодаря использованию мощных инструментальных средств время цикла разработки для всего проекта можно сократить;  **Недостатки** 1)Для больших проектов в RAD требуются существенные людские ресурсы (необходимо создать достаточное количество групп).2)RAD применима только для таких приложений, которые могут декомпозироваться на отдельные модули и в которых производительность не является критической величиной.3)RAD не применима в условиях высоких технических рисков (то есть при использовании новой технологии). | 1) в системах, которые поддаются моделированию (основанных на использовании компонентных объектов), а также в масштабируемых системах; 2) в системах, требования для которых в достаточной мере хорошо известны; 3) при выполнении проектов, разработка которых должна быть выполнена в сокращенные сроки; 4) когда затраты и соблюдение графика не являются самым важным вопросом; |
| 5. многопроходная | Представляет собой несколько итераций процесса построения прототипа программного продукта с добавлением на каждой следующей итерации новых функциональных возможностей или повышением эффективности программного продукта. |  | **Достоинства**  1) в начале разработки требуются средства только для разработки и реализации основных функций программного продукта; 2) после каждого инкремента получается функциональный продукт; 3) снижается риск неудачи и изменения требований; 4) улучшается понимание как разработчиками, так и пользователями программного продукта требований для более поздних итераций;  **Недостатки**  1) не предусмотрены итерации внутри каждого инкремента; 2) определение полной функциональности должно быть осуществлено в самом начале жизненного цикла разработки; 3) может возникнуть тенденция оттягивания решения трудных задач;4) общие затраты на создание программного продукта не будут снижены по сравнению с другими моделями; обязательным условием является наличие хорошего планирования и проектирования. | Многопроходная модель может быть применена, если большинство требований к программному продукту будут сформулированы заранее, а для выполнения проекта будет выделен большой период времени. |
| 6. прототипирования | Модель прототипитования позволяет создать прототип программного продукта до или в течение этапа составления требований к программному продукту. |  | Достоинства 1. Взаимодействие заказчика с разрабатываемой системой начинается на раннем этапе;2. Благодаря реакции заказчика на прототип сводится к минимуму число неточностей в требованиях;3. Снижается вероятность возникновения путаницы, искажения информации или недоразумений при определении требований к программному прдукту, что приводит к созданию более качественного программного продукта;4. В процессе разработки всегда можно учесть новые, даже неожиданные требования заказчика;5. Прототип представляет собой формальную спецификацию, воплощенную в программный продукт;  Недостатки  1. Решение сложных задач может отодвигаться на будущее;2. Заказчик может предпочесть получить прототип, а не законченную полную версию программного продукта;3. Прототипирование может неоправданно затянуться;4. Перед началом работы неизвестно, сколько итераций придется выполнить. |  |

2. Научиться осуществлять выбор стратегии и модели жизненного цикла для разработки конкретного проекта и обосновывать свой выбор. Процедура выбора модели жизненного цикла программных средств и систем

Для выбора подходящей к условиям конкретного проекта модели ЖЦ ПС и систем Институтом качества программного обеспечения SQI рекомендуется использовать специальную процедуру. Данная процедура базируется на применении четырех таблиц вопросов.

Каждая из табл. 3.1 – 3.4 представляет одну из категорий классификации проектов. Каждый из вопросов (строка в таблице) предназначен для классификации анализируемого проекта по определенному критерию категории. Столбцы данных таблиц соответствуют обобщенным моделям ЖЦ, фактически представляющим стратегии разработки ПС. При этом под RAD-моделью подразумевается независимая RAD-модель, не встроенная в другие модели жизненного цикла.

Рассматриваемая процедура состоит из следующей последовательности шагов:

*1-й шаг.* Проанализировать отличительные черты проекта по критериям категорий, представленным в виде вопросов.

*2-й шаг.* Ответить на вопросы по анализируемому проекту, отметив слова «да» или «нет» в соответствующих строках табл. 3.1 – 3.4. Если слов «да» или «нет» в строке несколько, необходимо отметить все из них (все «да» или все «нет»).

В качестве примера в табл. 3.1 выделены варианты ответов для проекта разработки сложного и критичного программного средства, требования к которому заранее не известны и будут уточняться по ходу разработки.

*3-й шаг.* Расположить по степени важности категории (таблицы) и/или критерии, относящиеся к каждой категории (вопросы внутри таблиц), относительно проекта, для которого выбирается модель ЖЦ.

*4-й шаг.* Выбрать из моделей (см. табл. 3.1 – 3.4) ту модель, которая соответствует столбцу с наибольшим количеством отмеченных ответов с учетом их степени важности (с наибольшим количеством отмеченных ответов в верхней части приоритетных таблиц). Выбранная модель ЖЦ является наиболее приемлемой для анализируемого проекта.

Таблица 3.1 – Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |

Продолжение таблицы 3.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4. | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |

Каскадная – 5, V-образная – 5, RAD – 4, инкрементная – 6, быстрого прототипирования – 2, эволюционная – 2.

На основе результатов заполнения табл. 3.1 наиболее подходящей является инкрементная модель.

Таблица 3.2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков  проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |

Каскадная – 4, V-образная – 5, RAD – 4, инкрементная – 5, быстрого прототипирования – 2, эволюционная – 1.

На основе результатов заполнения табл. 3.2 наиболее подходящими являются RAD и инкрементная модели.

Таблица 3.3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |

Продолжение таблицы 3.3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |

Каскадная – 3, V-образная – 3, RAD – 1, инкрементная – 1, быстрого прототипирования – 1, эволюционная – 3.

На основе результатов заполнения табл. 3.3 наиболее подходящими являются каскадная и эволюционная модели.

Таблица 3.4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| 2. | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| 3. | Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |

Продолжение таблицы 3.4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| 8. | Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| 9. | Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |

Каскадная – 5, V-образная – 7, RAD – 3, инкрементная – 5, быстрого прототипирования – 3, эволюционная – 5.

На основе результатов заполнения табл. 3.4 наиболее подходящей является V-образная модель.

Исходя из результатов заполнения табл. 3.1 – 3.4 (каскадная – 16, V-образная – 19, RAD – 10, инкрементная – 17, быстрого прототипирования – 9, эволюционная – 12) наиболее подходящей моделью, для разработки данного ПП, является V-образная модель.

**2.2 Инструменты разработки**

Для разработки данного проекта будет выбрана среда разработки WordPress, которая является наиболее актуальной средой для создания приложений данного типа.

Программа обладает удобным и понятным интерфейсом, а также полным набором необходимых функций для создания инсталляторов с минимальным размером, высокой степенью сжатия файлом и приятным интерфейсом.

Для создания пользовательского интерфейса будет использоваться Figma. Данное средство очень удобно бля проектирования ПИ и имеет все необходимые инструменты.

Также в создании проекта использовались такие приложения как:

1. Яндекс-браузер – нужен для нахождения информации и изображений.
2. Word 2016 – нужен для написания пояснительной записки.
3. Visio и XMind – нужны для создания графической части, а так же UML-диаграмм.

Параметры компьютера, который будет использован при разработке:

-Процессор: Intel(R) Pentium(R) CPU N3710 @ 1.60GHz;

-Объем оперативной памяти: 4.00 GB;

-Объем памяти на жестком диске: 1000 GB;

-Видеокарта: Intel(R) HD Graphics;

-ОС Windows 10 Pro.

**2.3 Разработка UML-диаграмм**

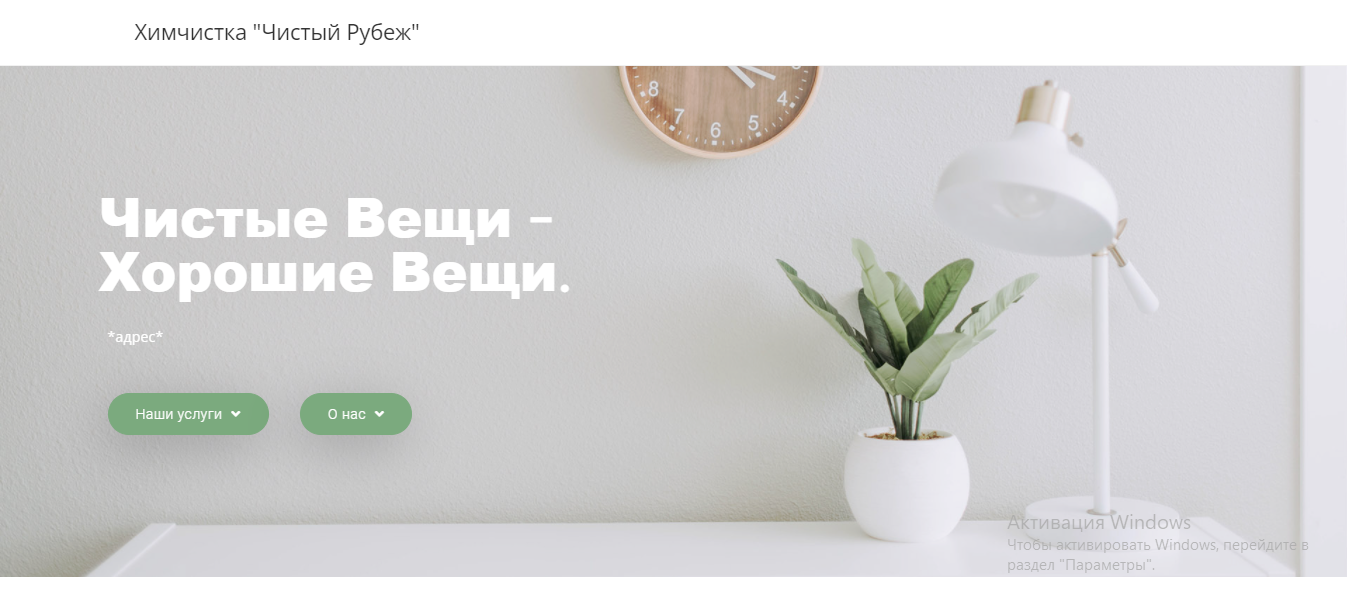
В ходе создания проекта требовалось создать две UML- диаграммы, что помогли бы понять определенные процессы самого интернет-ресурса. Этими диаграммами являются:

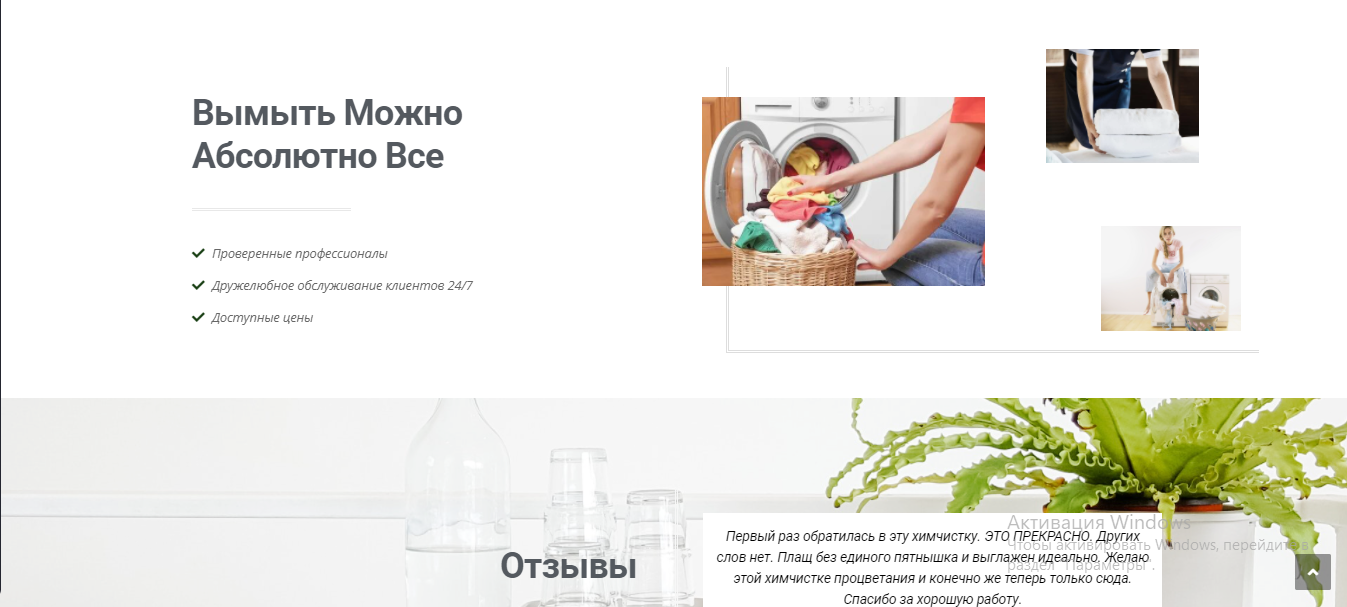
1 Диаграмма состояния;

1. Диаграмма последовательности;

**2.4 Разработка пользовательского интерфейса**

Перед началом разработки ПИ была разработана структура сайта. Структура сайта (карта сайта) — это логическая схема распределения страниц сайта по категориям / разделам (папкам).





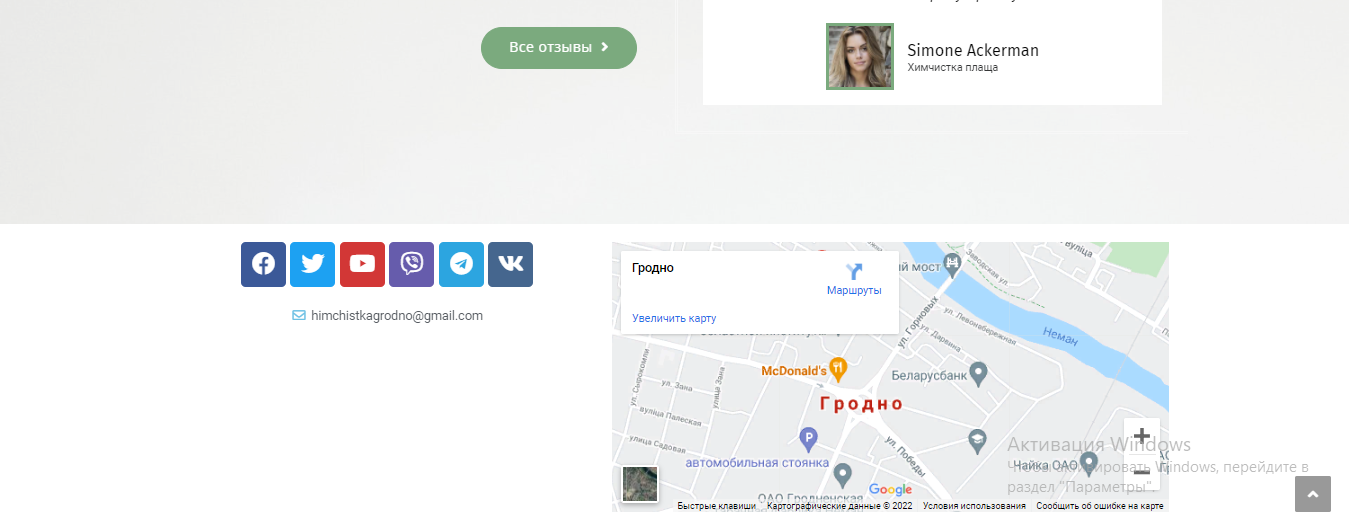


Рисунок 3 – Главная страницы

**3 Реализация**

**3.1 Руководство программиста**

Что бы создать сайт с помощью конструктора сайтов нужно найти хорошую площадку. Одна из них – WordPress.

Для начала с официального сайта скачивается WordPress, далее скачивается портативная серверная платформа OpenServer. В папке c OpenServer ищем папку domains и создаём пустую папку с названием своего сайта. Далее создаем базу данный и привязываем ее к своему сайту.

Далее входим в WordPress. Для начала устанавливаем плагины, которые облегчат работу:

1 Elementor – для удобного добавления и управления элементами сайта;

2 Envato Elements – готовые шаблоны сайтов;

3 User Registration – создание форм регистрации и авторизации;

4 Ultimate Member – создание личный кабинетов.

С помощью вкладки «Внешний вид» в панели меню можно выбрать тему и шаблоны сайта. Откроется окно с множеством шаблонов с различной тематикой, также присутствует строка поиска для более быстрого и удобного поиска.

После всех проделанных действий у вас появляется шаблон сайта, который можно изменять как вы хотите. Заходим во вкладку страницы, выбираем нужную страницу и нажимаем «Изменить». Появляется меню настроек сайта, где можно отключить автоматический подвал и так далее, а также можно выбрать устройство для которого редактируется сайт. Далее переходим в «Редактор Elementor». Далее создаем секции и перетаскиваем нужные элементы. Переходы между страницами можно осуществить с помощью меню вверху сайта, либо с помощью ссылок на элементах.

Страницы можно как сохранять в черновиках, так и осуществить публикацию. Рекомендуется сразу опубликовывать страницы, так как это дает возможность связывать страницы между собой, а также добавлять страницы в меню сайта для быстрого доступа к ним и дальнейшего редактирования их.

**4 Тестирование**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результатах тестирования предоставлен в таблице 5.

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Физический результат | Результат тестирования |
| 1 | 2 | 3 | 4 |
| Регистрация | Успешная регистрация | Неуспешная регистрация | Не выполнено |
| Авторизация | Успешная авторизация | Неуспешная авторизация | Не выполнено |
| Переход между страницами | Переход осуществляется между всеми возможными станицами | Переход осуществляется между всеми возможными станицами | Выполнено |

**5 Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Наименованием программного продукта является услуги химчистки. Назначение - для людей любого возраста, которые желают очистить одежду или вещи от загрязнений. Периодичность использования ПП - по мере необходимости.

**5.2 Запуск сайта**

Для того, чтобы запустить данный сайт на вашем компьютере, вам нужно лишь войти в любой из существующих браузеров, ввести ссылку сайта (http://cs86686.tmweb.ru/) в поисковик, и вы на самом сайте. Главное не забывайте, что вам нужно иметь хорошее интернет-соединение.

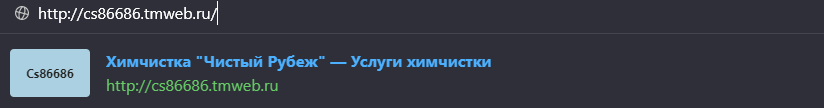


Рисунок 4 - Ввод ссылки в поисковик браузера

**5.3 Инструкции по работе с сайтом**

Для того, чтобы переходить по страницам сайта, вам требуется пройти авторизацию, для этого требуется нажать кнопку «Войти» на главной странице. Алгоритм перехода:

1 вы только вошли на сайт, оказавшись на Главной странице сайта;

2 выбираете на верхней панели кнопку для прохождения авторизации/регистрации;

3 вы перешли на страницу авторизации/регистрации;

4 далее вы переходите в личный кабинет, где сверху доступно меню.

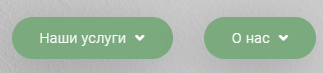


Рисунок 7 – Меню

**5.4 Завершение работы с сайтом**

Есть два способа завершить работу на нашем сайте: закрыть вкладку сайта в браузере или же закрыть сам браузер с помощью кнопки «Х» (Рисунок 10 – Рисунок 11).



Рисунок 10 – Закрытие вкладки



Рисунок 11 – Закрытие браузера

**Заключение**

Целью данного проекта являлась разработка сайта для услуг химчистки и стирки

В ходе реализации поставленной задачи был укреплён пройденный изученный материал по дисциплине «Технология разработки программного обеспечения», а также получено много дополнительной информации о её возможностях.

Созданный сайт предоставляет возможность использования услуг стирки и химчистки при необходимости.

После тщательного тестирования сайта были выявлены некоторые недоработки, которые были частично исправлены на стадии проектирования, и полностью исключены на стадии тестирования программы. В целом при реализации программы, были выполнены все условия, перечисленные в предыдущих разделах пояснительной записки. Таким образом, можно сказать, что программа была реализована вполне успешно.