Лабораторная работа № 1

«Методы и средства программной инженерии»

Вариант 1713

**Выполнила:**

студент группы Р32111

Линь Пэн

студентка группы P32151

Ван Минчжи

**Преподаватель:**

Цопа Евгений Алексеевич

**Цель работы:**

Составить список требований, предъявляемых к разрабатываемому веб-сайту (в соответствии с вариантом). Требования должны делиться на следующие категории:

1. Функциональные.
   1. Требования пользователей сайта.
   2. Требования владельцев сайта.
2. Нефункциональные.

Требования необходимо оформить в соответствии с шаблонами RUP (документ SRS - Software Requirements Specification). Для каждого из требований нужно указать его атрибуты (в соответствии с методологией RUP), а также оценить и аргументировать приблизительное количество часов, требующихся на реализацию этого требования.

Для функциональных требований нужно составить UML UseCase-диаграммы, описывающие реализующие их прецеденты использования.

**Вариант №1713:**

Свято-Троицкая Александро-Невская Лавра. Официальный сайт Свято-Троицкой Александро-Невской Лавры. История, Александр Невский, последние события, видео, фотографии, проповеди наместника - http://lavra.spb.ru/

**Документ SRS:**

1. Введение
   1. Цель: подготовить веб-сайт для Свято-Троицкая Александро-Невская Лавра. Веб-сайт должен отвечать следующим функциональным и нефункциональным требованиям.
   2. Соглашение о терминах:
      * Функциональность - функции сайта, просмотр расписания богослужений, презентаций и т.д.
      * Сервер - скрытая от пользователя часть сайта, которая обеспечивает работу всех представленных функций.
      * Веб-интерфейс - доступная пользователю, визуально удобная часть сайта.
      * PHP - язык программирования общего назначения. который будет использоваться для написания серверной части сайта.
      * База данных - объект, в котором хранятся данные, сохраненные пользователем. Расположена на сервере.
      * SQL - язык запросов для взаимодействия с базами данных.
      * PostgreSQL - один из типов баз данных, который будет использоваться в данном продукте.
      * HTML - один из инструментов, используемых для разработки веб-интерфейса сайта.
      * позволяет разместить все необходимые элементы на сайте (кнопки, меню, изображения, статьи и т.д.).
      * CSS - один из инструментов для разработки веб-интерфейса сайта.позволяет размещать красивые элементы сайта, используя HTML.
      * JavaScript (JS) - язык программирования, который будет использоваться при разработке для добавления интерактивных элементов на сайт (форм для отправки данных, ответов на рекламу, выделения выбранных элементов и т.д.), а также для соединения веб-интерфейса с серверной частью сайта.
      * AJAX — динамически обновляет страницу, снижая нагрузку на браузер.
      * Адаптивная верстка - способ создания веб-сайта, который одинаково легко отображается на всех устройствах, независимо от того, имеют ли они разные размеры экрана или разную ориентацию экрана. Таким образом, дизайн веб-сайта будет меняться в зависимости от поведения пользователя.
      * Масштабируемость - способность добавлять новые функции в проект в будущем, возможность и средства расширения текущего проекта.
      * Протокол – набор правил, определяющих обмен данными между двумя сторонами.
      * SSL – протокол передачи данных, позволяющий поддерживать безопасную связь.
      * Платежный шлюз Сбербанка – услуга ПАО Сбербанк, позволяющая безопасно делать пожертвования на сайте.
      * Логирование – технология, которая автоматически записывает и сохраняет информацию о событиях, происходящих в программе, облегчая отслеживание возможных ошибок и их причин.
   3. Предполагаемая аудитория: прихожане и паломники издалека. Конечно, есть и временные люди, которым интересна эта тема.
   4. ) Масштаб проекта: веб-сайт, открытый для всех, без регистрации, с описанием функций и содержания сайта, частично - краткое изложение событий прихожан, частично - описание услуг для паломников, частично - презентация монастыря для всех. Публикуется ряд конкретных статей с соответствующими фотографиями. Веб-сайт время от времени обновляется, например, новостями или изменениями адресов социальных сетей и поддержанием соответствующих статей.
   5. Ссылки на источники: https://habr.com/ru/post/52681/. Содержит правила подготовки программного обеспечения SRS.
2. Общее описание:
   1. Видение продукта: простой в использовании интерфейс с мягкими цветами. Сайт содержит подробное описание всех услуг, предлагаемых компанией.
   2. Функциональность продукта: имеет четкую и лаконичную главную страницу с фотографиями, связанными с аббатством. На главной странице размещены последние новости, а также описания и фотографии недавних событий.

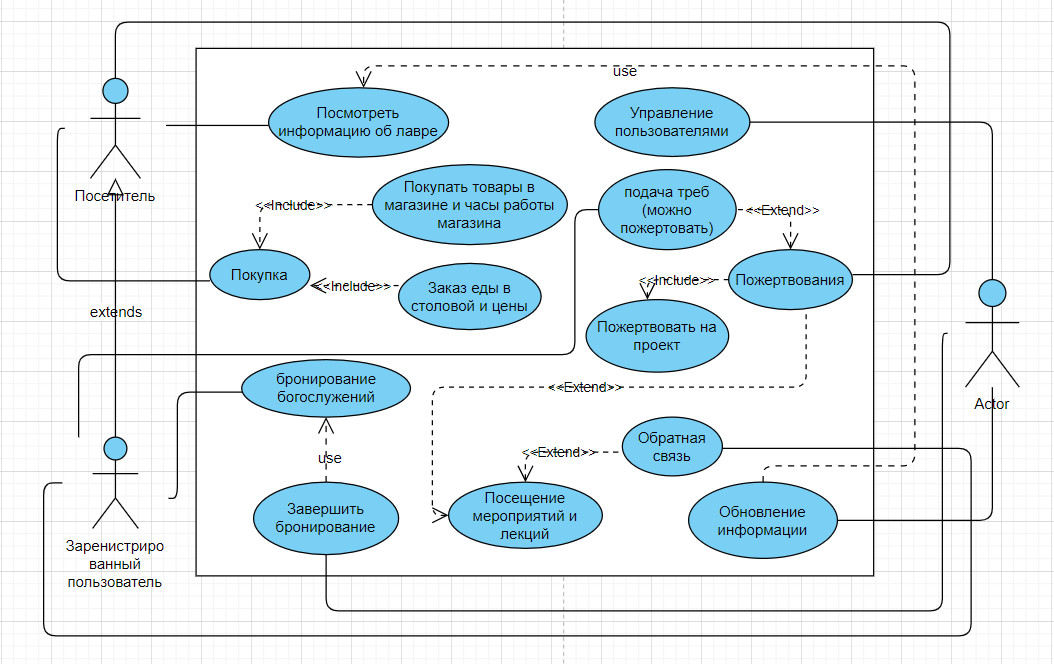
Подробно: расписание для верующих и некоторые требования. Для верующих - информация об аббатстве, гостинице, столовой, магазине и т.д. В дополнение к этому каждый может просмотреть некоторую информацию об аббатстве: рассказы, историю, персоналии и т.д., а также проекты об аббатстве. С аббатом также можно связаться через социальные сети.

* 1. Классы и характеристика пользователей: прихожане и паломники, приезжающие издалека. Конечно, есть и случайные люди, которые интересуются темой. В целом, доступ к нему может получить любой желающий.
  2. Среда функционирования проекта: браузер (настольный или мобильный), доступ в Интернет.
  3. Рамки, ограничения, правила и стандарты: обсуждение проекта с клиентом 2 раза в неделю, по четвергам и понедельникам (для удобства мы предполагаем, что работа над проектом начнется в понедельник). Общение между членами команды: полчаса в начале каждого рабочего дня. Ознакомьте сотрудников с этим документом, чтобы создать полное представление о разрабатываемом продукте. Соблюдение стандартов кодирования соответствующих языков программирования и первоначальное проектирование приложения всей командой.
  4. Документация для пользователей: существует меню, которое размещается на главной странице сайта в качестве навигации и представляет все функции сайта. Нет необходимости писать документацию для пользователей. Однако было отмечено, что необходимо обсудить с клиентом планы по поддержке проекта.
  5. Допущения и зависимости: После выполнения всех задач, описанных в данном документе, разрешается продолжить работу над проектом после написания соответствующего нового контракта. Разрешается предоставить системного администратора из команды разработчиков для поддержки проекта с регулярным обновлением технического обслуживания. Запланированных зависимостей нет.

1. Функциональные требования:
   1. Необходимо создать кнопку "меню". Это позволит пользователям легче просматривать и перемещаться по различным проектам.
   2. Ряд внешних ссылок на VK, Telegram и YouTube для более подробного ознакомления с аббатством.
   3. После каждой страницы есть кнопка: "Один клик назад на главную страницу". Нажмите эту кнопку, чтобы быстро вернуться на главную страницу.
   4. В левой части некоторых страниц указаны контактные телефоны, адреса и электронная почта. Если вы открыли страницу на мобильном телефоне, вы можете позвонить напрямую или перейти к навигационной системе мобильного телефона для автоматического поиска адреса.
   5. На некоторых страницах необходимо предусмотреть возможность загрузки файлов или перехода на другие страницы.
   6. Веб-мастеру необходимо обновлять статьи на сайте.
2. Требования к внешним интерфейсам:

Представлены в разделе видения продукта и согласованы с клиентом.

1. Нефункциональные требования.
   1. Создание веб-интерфейсов с использованием HTML, CSS, JS，DHTML.
   2. Использование PHP для создания back-end продукта.
   3. Использование библиотеки jQuery AJAX для обмена данными между бэк-эндом и веб-интерфейсом.
   4. Использование базы данных PostgreSQL для хранения данных пользователей для отдельных аккаунтов.
   5. Использование языка SQL для взаимодействия с базой данных.
   6. Добавление журналов PHP.
   7. Ежедневное резервное копирование всех данных и их хранение в отдельной базе данных.
   8. Запрос у клиента примерного количества групп клиентов, удвоение этого количества и обеспечение мощности сервера для поддержания работоспособности сайта при такой активности пользователей.
   9. Безопасность веб-сайта — предотвращение уязвимостей серверной системы веб-сайта, DDoS-атак, подделки веб-страницы, утечки данных веб-сайта и т. д. Быть атакованным нелегко, и его можно быстро восстановить после нападения.
   10. Стабильность веб-сайта: может позволить посещение многим пользователям и не подвержен сбоям; сбои могут быть быстро устранены
   11. Он может работать без сбоев в разных браузерах, системах и мобильных устройствах, и в нем не будет искаженных символов и т. д.
   12. Простота расширения, удобство для последующей оптимизации и расширения сайта
   13. Примечание. Веб-сайт должен соответствовать всем применимым законам и правилам.
   14. Обслуживание: Веб-сайт должен быть прост в обслуживании и обновлении.
   15. Юзабилити: веб-сайт должен быть простым в использовании и понятным.
2. UseCase-диаграмма для функциональных требований к продукту:



Как для прихожан, так и для паломников есть прямой доступ к главному экрану, знакомству с монастырем и некоторыми социальными проектами или пожертвованиями.

В отличие от паломников, прихожане могут также получить доступ к страницам для своих повседневных нужд (расписание и т.д.). Конечно, паломники также могут получить к ним доступ - все страницы сайта открыты для всех, но они мало интересны паломникам.

Для администратора требуется только простое обслуживание: изменить введение, изменить контактные данные и изменить содержание статей (при необходимости).

1. Риски.
   1. Использование пользователями старых версий браузера (например, из-за устаревших версий операционной системы на телефоне) и неотображение или неадаптивное отображение определенных элементов ⇒ тестирование на нескольких устройствах, от слабых и старых версий до современных, при разработке теста - некоторое время тратится на тестирование на старых телефонах, добавление альтернативных элементов, поддерживающих старые версии браузера.
   2. Не произойдет ли переход на страницу или загрузка файла, когда это требуется, или не произойдет ли загрузка. Требуется тестирование.
   3. На сайте слишком много повторяющихся функций и он недостаточно чистый, некоторые функции повторяются в меню

**Выводы.**

В этом эксперименте мы узнали о usecase и нарисовали диаграммы вариантов использования на языке uml. Проект был проанализирован в соответствии с сайтом, приведенным в теме, была написана SRS-документация, мы ознакомились с основами инструментов программной инженерии и некоторыми методами agile-разработки.