**1 Анализ задачи**

**1.1 Постановка задачи**

**1.1.1 Организационно-экономическая сущность задачи**

Наименование задачи: сайт «KinoJam» для бронирования билетов в кинотеатре

Цель разработки: создать сайт для бронирования билетов в кинотеатре таких как kinoteatr.megamag.by с упрощеным дизайном и интуитивно понятным интерфейсом;

Назначение: мой программный продукт разрабатываетсыя для людей любого возраста, желающих купить билет на необходимый сеанс удаленно.

Периодичность использования: по мере необходимости.

Обзор существующих аналогичных ПП: рассмотрим сайт megamag.by. На данном сайте есть такие возможности как поиск определенного фильма по названию, выбор кинотеатра в регионе, просмотр новостей (анонсы фильмов, новости сайта, специальные показы), список текущих фильмов, отложенные билеты пользователя, личный кабинет. Все функции будут присутствовать в новом ПП, но также будет добавлена функция оценки кинотеатра по отзывам пользователей, дополнительная информация по кинотеатру, способность пользователей оставлять отзывы о фильмах. На данном сайте присутсвует реклама, которой в новом сайте не будет.

**1.1.2 Функциональные требования**

Описание перечня функций и задач, которые должен выполнять будущий ПП:

Гость:

– просмотр страниц сайта;

– поиск фильма по названию;

– выбор региона;

– сортировка кинотеатров по оценкам пользователей;

– просмотр имеющихся сеансов;

– регистрация;

– чтение новостей;

– просмотр информации про определенный кинотеатр;

– чтение отзывов о фильмах от пользователей;

Пользователь: все те же функции, что может выполнять гость, но так же появляется:

– авторизация;

– бронирование и покупка билетов;

– список отложенных билетов;

– оценивание кинотеатра/фильма;

– создание и редактирование профиля;

– выход из личного кабинета;

**1.1.3 Описание процессов с входной, выходной и условно-постоянной информацией**

Вся информация, которой оперирует пользователь в процессе решения задачи подразделяется на:

– входную информацию;

– выходную информацию;

– условно-постоянную информацию.

Входной информацией выступают: название фильма, название области/города, заполнение полей регистрации, номер места, оценка или отзыв, вопросы пользователей, данные из кинотеатра.

Выходной информацией выступают: наличие данного фильма в кинотеатрах, дни и время сеанса; кинотеатры, которые есть в данной области/городе; список кинотеатров, день и время сеансов, новости, билет, новая оценка кинотеатра/новый отзыв о фильме, пользователь становится Гостем, обновленная база данных, новый фильм, обновленный сайт/профиль.

Условно-постоянной информацией выступают: данные сайта, название фильма, список кинотеатров и информация о них, список кинотеатров и информация о них, фильм, его оценка, дата и время сеансов; база данных всех кинотеатров, информация и данные сайта, все данные сайта , базы данных всех кинотеатров , данные кинотеатра.

**1.1.4 Эксплуатационные требования**

Требования к применению: помогает быстро купить билет и узнать о текущих новинках.

Требования к реализации: для реализации статических страниц и шаблонов должны использоваться языки HTML и CSS. Для реализации интерактивных элементов клиентской части должны использоваться языки JavaScript.

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 голубые-беловатые оттенки. Основные разделы сайта должны быть доступны с первой страницы. Грамотный пользовательский интерфейс. Сайт должен адаптироваться под компьютер, телефон и планшет.

Требования к хостингу:

– Поддерживает 1С-Битрикс, WordPress, Joomla!, Drupal и любые другие CMS

– Круглосуточный мониторинг серверов 24/7: сбои устраняются раньше, чем их можно заметить

– Объем дискового пространство 8Гб,

– Эффективная защита от спама и взлома

– Базы данных хранятся на SSD-дисках.

– Максимальный размер базы данных - 2 Гб

– Все данные проходят через процедуру ежедневного резервного копирования. Копии хранятся в течение 20 дней.

– Поддержка MySQL, Python, PHP, Ruby, Node.js

– Стоимость хостинга составит 13,77р в месяц (165,24 за год)

– Стоимость домена Леки.бел 25,20 за год

– https://domain.by/hosting-order/#tariffs

**1.2 Диаграмма вариантов использования**

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования.

Актером или действующим лицом является любая сущность, взаимодействующая с системой извне. Это может быть как живое существо, так и любая друга система, которая может служить источником воздействия на моделируемую систему так, как определяет сам разработчик.

Вариант использования является стандартным языком UML и применяется для спецификаций общих особенностей системы и любой другой сущности. Отдельные варианты использования обозначаются на диаграмме эллипсом, в котором содержится его краткое название.

Отношение ассоциации является главным понятием языка UML и используется при построении всех графических моделей. Оно служит для обозначения роли актера и отдельном варианте использования. На диаграмме отношение ассоциации обозначается сплошной линией между актером и вариантом использования.

Для отображения взаимосвязи экземпляра отдельного варианта использования с более общим вариантом, используется отношение расширения, обозначаемое направленной пунктирной линией со стрелкой от исходного варианта. Данная линия помечается ключевым словом «extend».

Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Данная линия помечается ключевым словом «include».

Определяя для выбранного актера варианты использования и устанавливая отношения между вариантами использования, получается полная диаграмма вариантов использования. Её можно увидеть в Приложении А.

**2 Проектирование**

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

Для разработки программного продукта следует выбрать стратегию разработки и модель жизненного цикла. Выбрается посредством составления таблиц:

Таблица 1 – Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |

Вычисления: 2 за каскадную, 2 за V-образную, 5 за RAD, 3 за инкрементную, 5 за быстрого прототипирования и 5 за эволюционную.

Итог: Подходят модель быстрого прототипирования, RAD и эволюционная.

Таблица 2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории команды разработчиков проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |

Вычисления: 5 за каскадную, 5 за V-образную, 4 за RAD, 5 за инкрементную, 2 за быстрого прототипирования и 1 за эволюционную.

Итог: Подходит инкрементная, каскадная,V-образная модели.

Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 1 за каскадную, 1 за V-образную, 1 за RAD, 2 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: Подходит модель быстрого прототипирования и эволюционная.

Таблица 4 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 2 за каскадную, 3 за V-образную, 3 за RAD, 7 за инкрементную, 7 за быстрого прототипирования и 9 за эволюционную.

Общий итог: в итоге заполнения таблиц – наиболее подходящей является инкрементная модель.

Итог: подходит эволюционная модель.

Мы выбрали данную модель ЖЦ ПО, потому что наш проект невозможно было бы сделать с использованием любой другой модели из-за постоянного изменения постановки задачи. А также из-за постоянной конкуренции, необходимо вносить множество изменений. При этом мы будем работать с новыми технологиями. Из этого следует, что выбор любой другой модели ЖЦ ПО нам недоступен.

**2.2 Инструменты разработки**

Для разработки UML-диаграмм использовалось приложение «Draw.io», так как данное приложение имеет удобный и понятный интерфейс для составления диаграмм.

Для UX/UI дизайна использовалась «Figma», так как в данном приложении удобно работать с компонентами — заранее подготовленными элементами дизайна (кнопки, формы и т. д.), изменения которых распространяются на все его копии.

Для составления документации использовался «Microsoft Word», так как эта программа наиболее подходит для выполнения данной задачи.

Для разработки данного проекта я выбрал JavaScript.

Java Script — один из самых популярных языков программирования, который может использоваться как для работы в роли бэкенда, так и для фронтенда. Однако, роль Java в каждом случае может быть значительно различной.

Когда речь идет о бэкенде, Java Script широко используется для создания веб-сайтов.

Кроме того, Java широко используется в бэкенде благодаря своей способности к масштабированию. Если ваш веб-сервис растет, вы можете просто добавлять новые серверы, чтобы обрабатывать все больше запросов.

Для разработки на Java, HTML, CSS вам понадобятся следующие инструменты:

• Visual Studio Code - это бесплатная и мощная среда разработки с открытым

исходным кодом, которая поддерживает множество языков и расширений.

HTML:

Язык гипертекстовой разметки. Используется для описания веб-страниц с помощью обычного текста.

Его функционал — это создание и структурирование текста на сайте.

Разделы, заголовки, ссылки и абзацы — все это относится к HTML.

CSS:

Каскадные таблицы стилей. Применяются в сочетании с HTML и определяют внешний вид сайта. CSS отвечает за размер, цвет и положение всех элементов страницы, написанной на HTML.

• Документация по Java Script которую вы можете найти на сайте [learn.microsoft.com] или в [Википедии].

My SQL и PHP для создания БД.

• Обучающие материалы, такие как курсы, книги, видео и статьи, которые помогут вам изучить основы и продвинутые темы Java. Вы можете найти некоторые из них на сайте [learn.microsoft.com] или в [Stack Overflow].

• Библиотеки и фреймворки, которые расширяют функциональность Java,HTML,CSS, предоставляя готовые решения для часто встречающихся задач.

Вы можете найти их в реестре NuGet или на GitHub.

Разработка проекта будет происходить на компьютере со следующими параметрами:

– процессор Ryzen 5;

– объем ОЗУ 8 GB;

– объем места на HDD: 256 GB;

– ОС: Windows 10 Домашняя.

**2.3 Разработка UML-диаграмм**

В первом разделе «Постановка задачи» были описаны функциональные требования к проекту. На их основе построена диаграмму последовательности.

Диаграмма последовательности – это диаграмма, предназначенная для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними.

Графические диаграммы последовательности имеют два представления. Одно – слева направо – в виде вертикальных линий, которые соответствуют линии жизни отдельного участка взаимодействия. Второе – вертикальная временная ось, направленная сверху вниз.

Диаграмма последовательности представлена в Приложении А.

**2.4 Разработка пользовательского интерфейса**

Важным элементом проектирования данного программного продукта является описание внешнего интерфейса разрабатываемого программного продукта. Для разработки визуального дизайн использовались сдержанные цвета для удобства использования программного продукта. Разработанная структура приложения расположена в Приложении А.

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. ПП должен позволить пользователю решать задачи, затрачивая значительно меньше усилий, чем при работе с разрозненными объектами. Все исходные данные будут разделены на несколько групп.

Диаграмма деятельности представлена в Приложении А.

UX/UI макеты представлены в Приложении А.

**2.5 Тесты на использование**

В ходе разработки веб-приложения были составлены тесты, которые необходимо выполнить в дальнейшем. Тесты составлены таким образом, чтобы предусмотреть максимальное количество возможных действий. Тесты описаны в таблице в Приложении Б.

1. **Реализация**

**3.1 Руководство программиста**

В этой части кода HTML описаны кнопки навигации, иконки соц сетей и иконка личного кабинета (рисунок 1).

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание**

Рисунок 1 – код HTML

Здесь описаны стили классов, которые мы использовали в HTML (рисунок 2).

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

рисунок 2 – код CSS

Следующий код описывает функцию создания афишы одного фильма (рисунок 3).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

рисунок 3 – код JS

В этом коде описывали регистрацию и занос в базу данных (рисунок 4).

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

рисунок 4 – код PHP

Здесь описывается код со страницы профиля (рисунок 5).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

рисунок 5 – код PHP файла profile.php

**4 Тестирование**

Тестирование программного обеспечения было осуществлено в соответствии с тестами на использование, описаными в разделе «Тесты на использование».

|  |  |
| --- | --- |
| Таблица 8 – Тестирование функций | |
| № | Статус |
| Т1 | Выполнено успешно |
| Т2 | Выполнено успешно |
| Т3 | Выполнено успешно |
| Т4 | Выполнено успешно |
| Т5 | Выполнено успешно |
| Т6 | Выполнено успешно |
| Т7 | Выполнено успешно |

В результате проведения тестирования программного продукта выяснилось, что все раннее оговоренные функции были разработаны, а также протестированы. Тесты показали, что все функции работают правильно.

**5 Руководство пользователя**

При запуске сайта нас встречает главная страница. (рисунок 1).

Изображение выглядит как текст, Человеческое лицо, снимок экрана, человек

Автоматически созданное описание

Рисунок 1 – Главная страница

Можете выбрать день, в который вы хотите посетить кинотеатр,а также сам кинотеатр (рисунок 2,3).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2 – Выбор кинотеатра

Изображение выглядит как пожар, камин, жара, снимок экрана

Автоматически созданное описание

Рисунок 3 – День выбран

После выбора дня и кинотеатра, нажмите «15 руб» и откроется окно с подробной информацией о фильме, а так же время и цена (рисунок 4).

Изображение выглядит как текст, меню, Быстрое питание, снимок экрана

Автоматически созданное описание

Рисунок 4 – Информация о фильме

При нажатии «Кино», вас скролит к данному подпункту(рисунки 5,6), при нажатии «Новости» — к новостям(рисунок 7), при нажатии «О нас» — к информации о кинотеатре(рисунок 8), при нажатии «Регистрации» — переводит к странице с регистрацией(рисунок 9).

Изображение выглядит как текст, Шрифт, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 5 – Кнопки навигации

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 6 – Сегодня в кино

Изображение выглядит как снимок экрана, одежда, окно, в помещении

Автоматически созданное описание

Рисунок 7 – Новости

Изображение выглядит как текст, снимок экрана, небо

Автоматически созданное описание

Рисунок 8 – О нас

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 9 – Регистрация

При нажатии следующих иконок (рисунок 10), нас переводит на соц сети сайта (рисунок 11) и личный кабинет (или авторизация) (рисунок 12).

Изображение выглядит как Графика, снимок экрана, логотип, Шрифт

Автоматически созданное описание

Рисунок 10 – Иконки

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 11 – Соц сеть сайта

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 12.1 – Авторизация

Изображение выглядит как текст, снимок экрана, Человеческое лицо, графический дизайн

Автоматически созданное описание

Рисунок 12.2 – Личный кабинет

Из личного кабинета можно выйти по кнопке «Выйти» (рисунок 13).

Изображение выглядит как текст, Шрифт, Графика, графический дизайн

Автоматически созданное описание

Рисунок 13– Кнопка «Выйти»

При регистрации гостю необходимо ввести ФИО, пол,e-mail, телефон и пароль, после чего он записывается в базу данных (рисунок 14).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 14.1 – Регистрация

Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание

Рисунок 14.2 – База данных

Приложение А

Изображение выглядит как диаграмма, План, Технический чертеж, схематичный

Автоматически созданное описание

Рисунок А.1 – Структура сайта

Изображение выглядит как текст, диаграмма, Параллельный, План

Автоматически созданное описание

Рисунок А.2 – Функциональная модель

Изображение выглядит как текст, диаграмма, План, схематичный

Автоматически созданное описание

Рисунок А.3 – Модель данных

Изображение выглядит как диаграмма, рисунок, зарисовка, текст

Автоматически созданное описание

Рисунок А.4 – Диаграмма видов использования

Изображение выглядит как текст, диаграмма, линия, Параллельный

Автоматически созданное описание

Рисунок А.5 – Диаграмма последовательности

Изображение выглядит как зарисовка, диаграмма, рисунок, оригами

Автоматически созданное описание

Рисунок А.6 – Диаграмма деятельности

Изображение выглядит как диаграмма, План, Технический чертеж, линия

Автоматически созданное описание

Рисунок А.7 – Диаграмма классов

Изображение выглядит как диаграмма, линия, зарисовка, рисунок

Автоматически созданное описание

Рисунок А.8 – Диаграмма объектов

Изображение выглядит как текст, Человеческое лицо, снимок экрана

Автоматически созданное описание

Рисунок А.9 – UI макет формы

**Изображение выглядит как снимок экрана, текст, дизайн, окно

Автоматически созданное описание**

Рисунок А.10 – UX макет формы

Изображение выглядит как текст, программное обеспечение, веб-страница, снимок экрана

Автоматически созданное описание

Рисунок А.11 – макет регистрации

Изображение выглядит как снимок экрана, Прямоугольник, текст, прямоугольный

Автоматически созданное описание

Рисунок А.12 – макет бронирования

Изображение выглядит как текст, снимок экрана, программное обеспечение, диаграмма

Автоматически созданное описание

Рисунок А.13 – макет профиля

Приложение Б

Тесты на использование

Таблица Б – Тесты на использование

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название теста | Действие | Ожидаемый результат | Фактический результат | Статус |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Кнопки соц сетей | Нажатие на иконку любой соц сети | Открытие выбранной страницы сайта в соц сети | Открытие выбранной страницы сайта в соц сети | Выполнено успешно |
| 2 | Регистрация | Нажатие на кнопку «Регистрация» | Открывается новая страница с регистрацией | Открывается новая страница с регистрацией | Выполнено успешно |
| 3 | Авторизация | Нажатие на иконку личного кабинета | Открывается страница с авторизацией | Открывается страница с авторизацией | Выполнено успешно |
| 4 | Кнопка “Выход” | Нажатие на кнопку «выход»(нужно быть авторизованным) | Выход из личного кабинета, переход на главную страницу | Выход из личного кабинета, переход на главную страницу | Выполнено успешно |
| 5 | Проверка адаптивности сайта | Уменьшить размер экрана | Все элементы сайта уменьшаются пропорционально размеру экрану | Все элементы сайта уменьшаются пропорционально размеру экрану | Выполнено успешно |
| 6 | Данные пользователя | Авторизоваться | Запуск страницы профиля с данными пользователя(из базы данных) | Запуск страницы профиля с данными пользователя(из базы данных) | Выполнено успешно |