

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук
Кафедра информационных систем

Справочник достопримечательностей
Курсовой проект
по дисциплине Технологии программирования

09.03.02 Информационные системы и технологии

Обучающийся _____ *Глущенко В.С., 3 курс, д/о*
Обучающийся _____ *Дуров М.А., 3 курс, д/о*
Обучающийся _____ *Шипилов А.В., 3 курс, д/о*
Руководитель _____ *Тарасов В. С.*

Воронеж 2020

Оглавление

1. Введение	3
2. Постановка задачи	4
3. Цели создания системы	5
4. Анализ предметной области	6
5. Архитектура Справочника	10
6. Диаграммы	13
6.1. Диаграмма прецедентов	13
6.2. IDEF0 диаграмма	13
6.3. Диаграмма классов	15
6.4. Диаграмма объектов	16
6.5. Диаграмма последовательностей	16
6.6. Диаграмма коопераций	19
6.7. Диаграмма состояний	20
6.8. Диаграмма активностей	21
6.9. Диаграмма развертывания	24
7. Анализ ключевых сценариев	26
8. Реализация приложения	28
8.1. Выбор используемых технологий	28
8.2. Интерфейс	29
8.3. Система аналитики	38
9. Тестирование	41
10. Заключение	46
11. Перечень принятых определений и сокращений	47
Приложение А	49
Приложение Б	52

1. Введение

В современном мире у человека обычно все меньше времени на усвоение все большего количества информации. Именно поэтому так востребованы сервисы, в которых собраны воедино и сгруппированы какие-либо сведения об определенной предметной области. Не стала исключением и область достопримечательностей. Ведь это такие объекты, которые могут помочь человеку узнать много интересных фактов об истории своего родного края или страны.

Использование веб-сайта, предоставляющего информацию о достопримечательностях, дает следующие преимущества:

- Возможность просмотреть достопримечательности, не выходя из дома.
- Возможность узнать некоторые интересные сведения о достопримечательности.
- Узнать адрес понравившейся достопримечательности для того, чтобы посмотреть на нее вживую.
- Спланировать предстоящую поездку, выбрав для себя интересующие достопримечательности.

2. Постановка задачи

Целью данной работы является создание веб-сайта для просмотра информации о достопримечательностях и их оценки.

Конечным пользователем сайта будет являться человек, вне зависимости от рода его занятий.

К разрабатываемому продукту предъявляются следующие основные требования:

- Предоставление Пользователю возможности просмотра Достопримечательности
- Предоставление Авторизованному пользователю возможности на оформление заявки о добавлении достопримечательности.
- Предоставление возможности оценки достопримечательности, загруженной на сайт для АП.
- Возможность фильтрации, сортировки и поиска достопримечательностей, имеющихся на сайте.

Необходимо проанализировать уже существующие решения для того, чтобы выделить в них какие-либо недочеты и учесть это в разработке своей системы. А также для того, чтобы подметить какие-либо удачные вещи и внедрить их в свою систему.

3. Цели создания системы

Справочник создается с целью:

- создания реестра достопримечательностей России.
- упрощение получения информации о них Пользователями.
- повышения уровня заинтересованности
достопримечательностями России.

4. Анализ предметной области

Сначала был проведен анонимный опрос для выявления уровня осведомленности людей о достопримечательностях России. Опрос происходил путем заполнения бланка ответов на физическом носителе, то есть листе бумаги. Сам бланк представлен ниже:

Бланк опроса об осведомленности о достопримечательностях России

Ваш возраст:	
Как вы оцениваете свои знания о достопримечательностях России? (по 5-бальной системе)	
Как вы оцениваете свои знания о достопримечательностях Воронежа? (по 5-бальной системе)	
Знаете ли вы, в каком городе находится Исаакиевский собор?	
Знаете ли вы, в каком городе находится Мечеть <u>Кул Шариф</u> ?	
Знаете ли вы, на какой улице находится памятник Платонову в Воронеже?	
Знаете ли вы, на какой улице находится Воронежский Краеведческий музей?	
Хотели бы вы знать больше о достопримечательностях России?	

Рисунок 1 - Бланк осведомленности

Этот способ опроса был выбран в связи с тем, что было необходимо охватить наибольший возрастной круг лиц, а, как известно, люди старшего поколения в нашей стране с нежеланием заполняют различные электронные анкеты.

Благодаря данному решению удалось охватить широкую аудиторию в возрасте от 14 до 70 лет. Но, в таком случае, удалось опросить лишь тех, кто проживает в городе Воронеже. Поэтому сам бланк и был скорректирован в расчете на жителей именно этого города.

Оценка осведомленности рассчитывалась следующим образом: сначала человек самостоятельно оценивал свои знания о достопримечательностях России и Воронежа по пятибалльной системе, после чего отвечал на 4 вопроса, по 2 на каждую из выделенных категорий. При ответе на 0 из 2 вопросов респондент получал минус 1 балл к собственной оценке, 1 из 2 – балл не менялся, 2 из 2 – плюс 1 балл к собственной оценке. Больше 5 баллов получить было невозможно.

После чего рассчитывалась среднее из 2 оценок по категориям. Оценки ниже 4 приравнивались к слабой осведомленности, в то время как 4 и 5 свидетельствовали о хорошей осведомленности. Сводная таблица оценок представлена ниже:

Оценка	Кол-во респондентов
На оценку ниже 3	3
На оценку 3	8
На оценку 4	12
На оценку 5	7

Таблица 1 - Сводная статистика

Путем вычислений было получено, что среди 30 опрошенных лишь 19 человек получили оценку выше 3, то есть лишь 63 процента опрошенных. Также, важно отметить, что лишь 20 процентов не хотели бы знать больше о достопримечательностях России.

В связи с этим фактом и было принято решение о создании Справочника.

Сначала были рассмотрены аналогичные решения, существующие в интернете. Были взяты 2 сайта, размещенные по адресам: <https://autotravel.ru/> и <https://www.tripadvisor.ru/>

С <https://autotravel.ru/> было взято удачное наполнение и размещение информации на странице Достопримечательности. Но стоит отметить, что данный сервис в большей степени ориентирован на поиск готовых туристических маршрутов, поэтому поиск какой-то конкретной достопримечательности является затруднительным, да и с интерфейсом данного сайта разберется далеко не каждый пользователь.

На этом сервисе реализовано добавление Достопримечательностей, но это можно сделать лишь с помощью Форума или почты, так как отдельная форма для данной функции не реализована.

Сайт <https://www.tripadvisor.ru/> представляет удобный способ фильтрации и сортировки Достопримечательностей, которые были взяты за основу для нашего Справочника. Страница Достопримечательности же является перегруженной различной информацией о находящихся поблизости ресторанах и отелях, так как именно это и является основным направлением сайта, что, очевидно, не совпадает с целями нашего Справочника, вследствие чего было принято решение отказаться от реализации подобных функций в рамках создания Системы.

На основе анализа существующих решений, и, принимая во внимание направленность Системы, были сделаны следующие выводы:

- Интерфейс системы должен быть рассчитан на пользователей, не имеющих специальных технических знаний и навыков в области компьютерной техники, и быть легко осваиваем ими.

- Основное преимущество нашей системы, выделяющее ее на фоне остальных решений – простота добавления Достопримечательности пользователем. В связи с тем, что пользователь может использовать, например, ненормативную лексику, было принято решение, что перед добавлением Достопримечательности проходят проверку модератором.

- Еще одно отличное решение – добавить оценки для каждой достопримечательности, которые будут формироваться как среднее арифметическое всех оценок, поставленных Достопримечательности.

- Так как Достопримечательностей очень много, то отличным решением будет выбрать какую-либо страну, достопримечательности которой будут рассмотрены. В данной работе будут рассматриваться достопримечательности России и возможность их систематизации, так как выяснилось, что люди в нашей стране слабо осведомлены о них.

- Будут выделены следующие основные виды достопримечательностей:

1. Архитектура и памятники.
2. Религиозные объекты.
3. Природные объекты.
4. Развлекательные объекты.
5. Исторические объекты.
6. Музеи.

- Для удобства Пользователей будут реализованы сортировки по алфавиту и оценке, а также поиск по названию.

5. Архитектура Справочника

Для реализации данного приложения используется шаблон проектирования MVT (Model View Template):

Данный паттерн проектирования очень близок к классическому MVC (Model View Contoller). Основное различие в том, что в MVT Django сам заботится о части контроллера, а именно о программном коде, контролирующем взаимодействие между моделью и представлением.

Основная информация об MVT:

- документация Django определяет модель (model) как «источник информации о данных, в которых содержатся ключевые поля и поведение данных».
- Обычно одна модель указывает на одну таблицу в базе данных.
- Модель отвечает за бизнес-логику, методы, свойства и другие элементы, связанные с манипуляцией данными.
- Также модели позволяют разработчикам создавать, читать, обновлять и удалять объекты в базе данных.

Представление (view) решает три задачи: принимает HTTP-запросы, реализует бизнес-логику, определённую методами и свойствами, отправляет HTTP-ответ в ответ на запросы. То есть представление получает данные от модели и предоставляет шаблонам (templates) доступ к этим данным или предварительно обрабатывает данные и затем предоставляет к ним доступ шаблонам.

В Django реализована мощная система шаблонов и собственный язык разметки. Шаблоны представляют собой файлы с HTML-кодом, с помощью которого отображаются данные. Содержимое файлов может быть статическим

или динамическим. Шаблоны не содержат бизнес-логики. Поэтому они только отображают данные.

Другие шаблоны проектирования:

1. MVVM (Model-View-View-Model)

MVVM больше всего подходит под случай, когда нужно отображать большое количество данных без предварительной обработки на стороне view.

Почему мы не выбрали MVVM:

Во-первых, MVVM заставляет работать с view двумя путями: через databinding и через методы View.

Во-вторых, нужно использование DatabindingLibrary, что требует времени на освоение этой библиотеки

2. MVP (Model View Presenter)

MVP — шаблон проектирования, производный от MVC, который используется в основном для построения пользовательского интерфейса. Элемент Presenter в данном шаблоне берёт на себя функциональность посредника (аналогично контроллеру в MVC) и отвечает за управление событиями пользовательского интерфейса так же, как в других шаблонах обычно отвечает представление.

Почему мы не выбрали MVP:

Во-первых, раздутый интерфейс View.

Если экран имеет много элементов и состояний, это приводит к тому, что в интерфейс View добавляется слишком много методов. Presenter будет использовать каждый метод для обновления маленькой части UI экрана.

Во-вторых, сложно сохранять состояние.

В правильной реализации MVP состояние хранится в Model. Пользователь делает действие - Presenter получает обновленную Model - Presenter обновляет View.

6. Диаграммы

6.1. Диаграмма прецедентов

Как мы видим на диаграмме, в нашей Системе будет существовать 3 вида пользователей: Модератор, Неавторизованный пользователь, АП, каждый из которых способен взаимодействовать со Справочником различными способами.

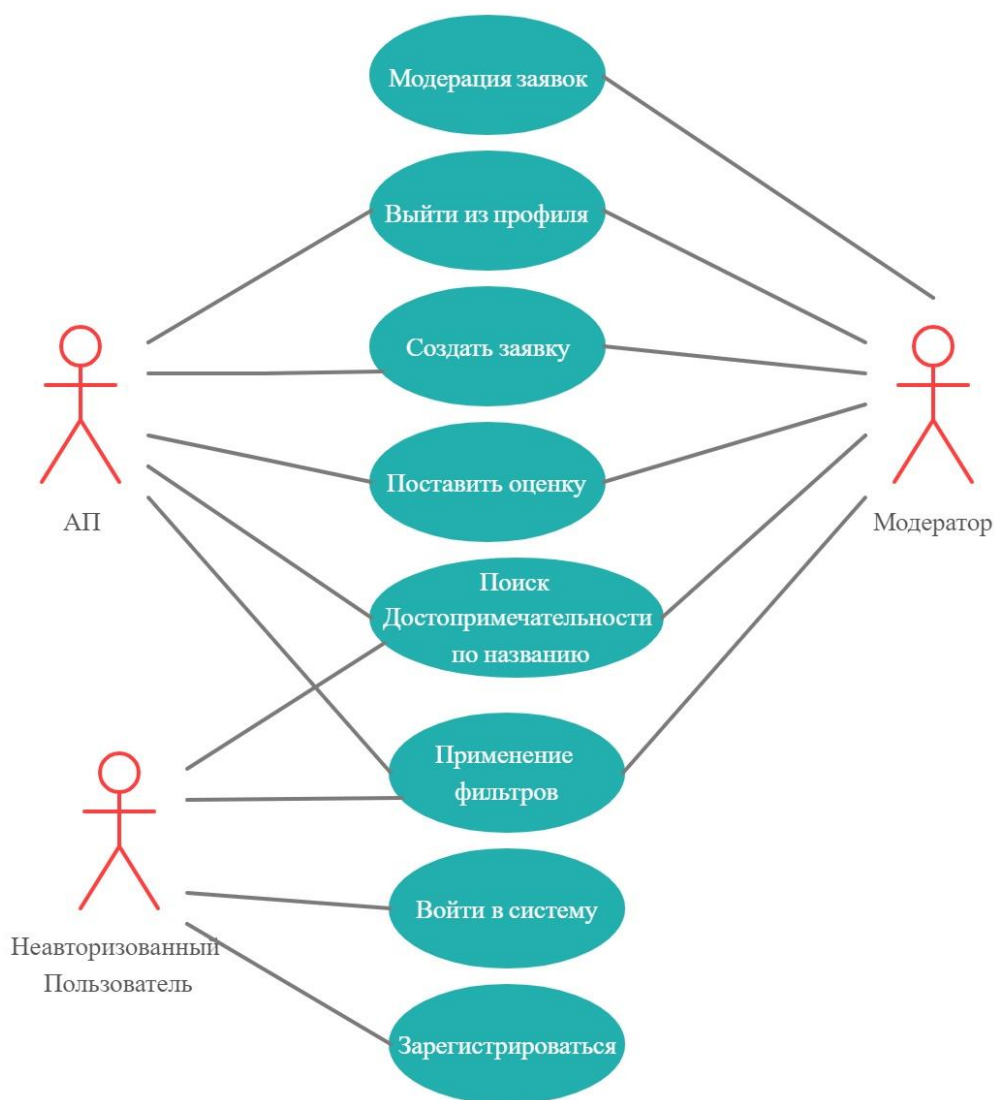


Рисунок 2 - Диаграмма прецедентов

6.2. IDEF0 диаграмма

На данной диаграммы представлено видение добавления Достопримечательностей в Справочник.

В блок «Заполнить заявку» на вход поступает информация о достопримечательности из внешних источников, Управление осуществляет Авторизованный пользователь, Механизм – Правила заполнения заявки. На выходе получаем Заявку.

После чего эта Заявка поступает на вход в блок «Принять заявку». Управление осуществляет Модератор, Механизмы - Правила заполнения заявки и Информация из сети Интернет. Выход – новая Достопримечательность.

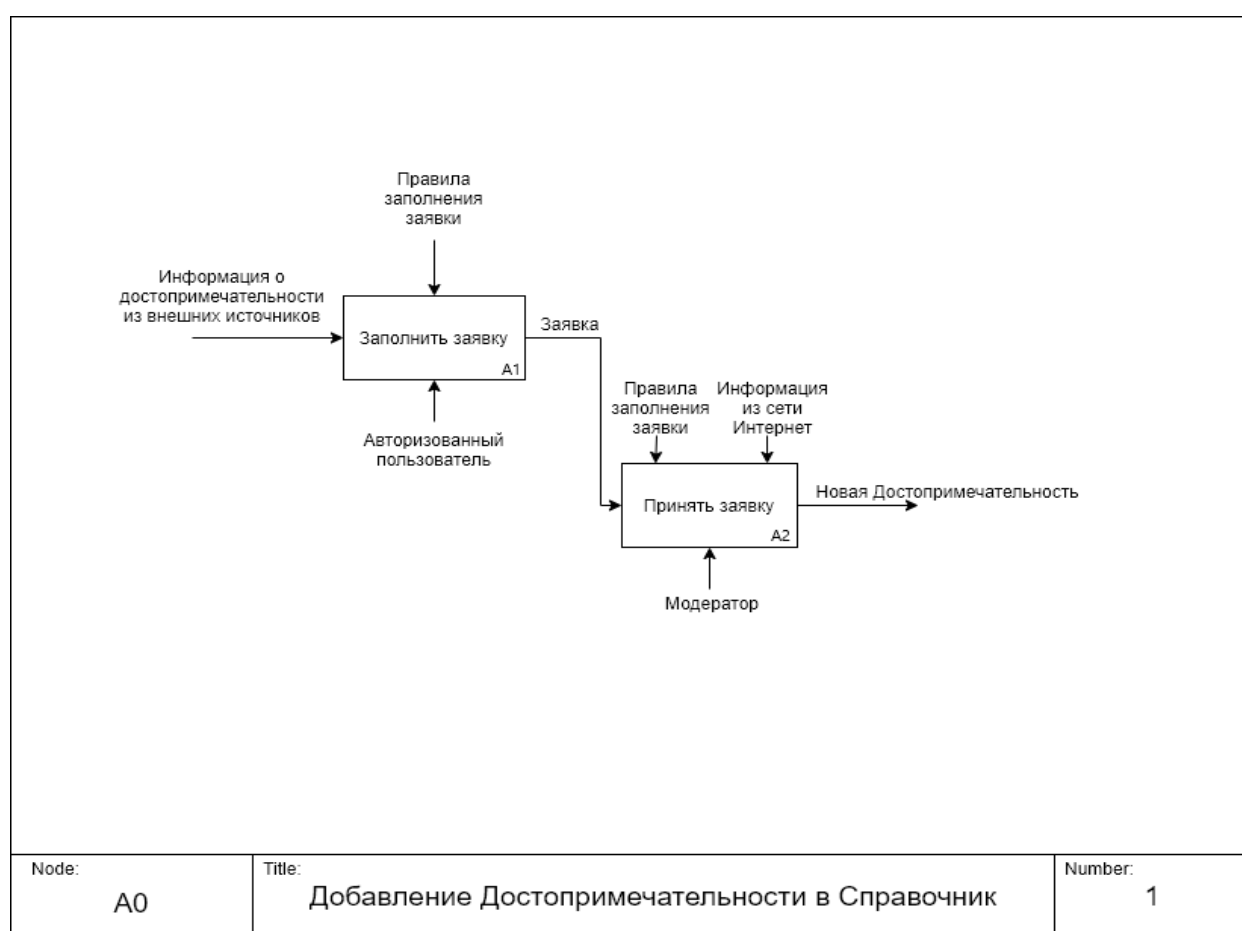


Рисунок 3 - IDEF0

6.3. Диаграмма классов

Диаграмма классов – это структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.

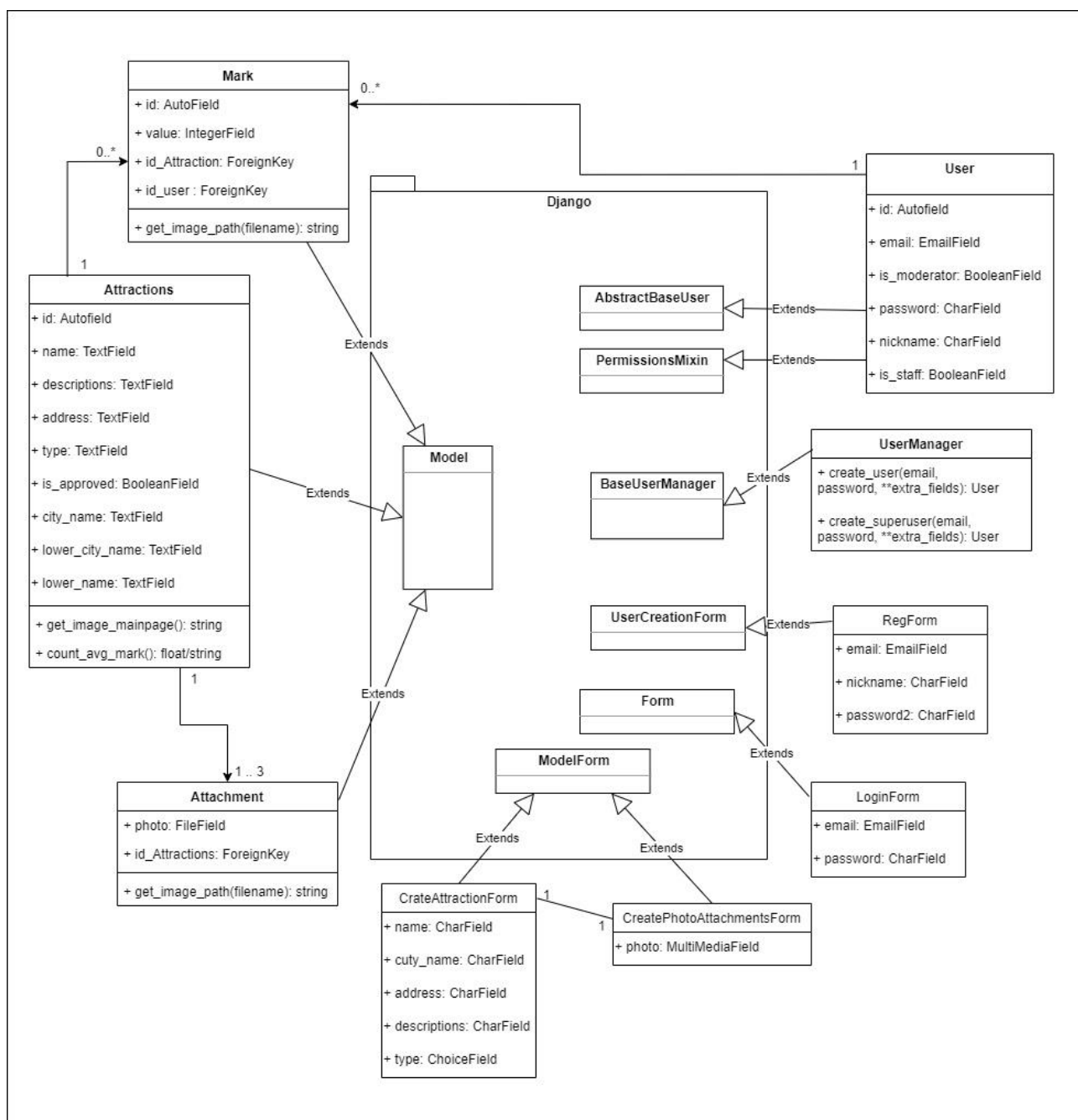


Рисунок 4 - Диаграмма классов

6.4. Диаграмма объектов

На диаграмме объектов отображаются экземпляры классов (объекты) системы с указанием текущих значений их атрибутов и связей между объектами.

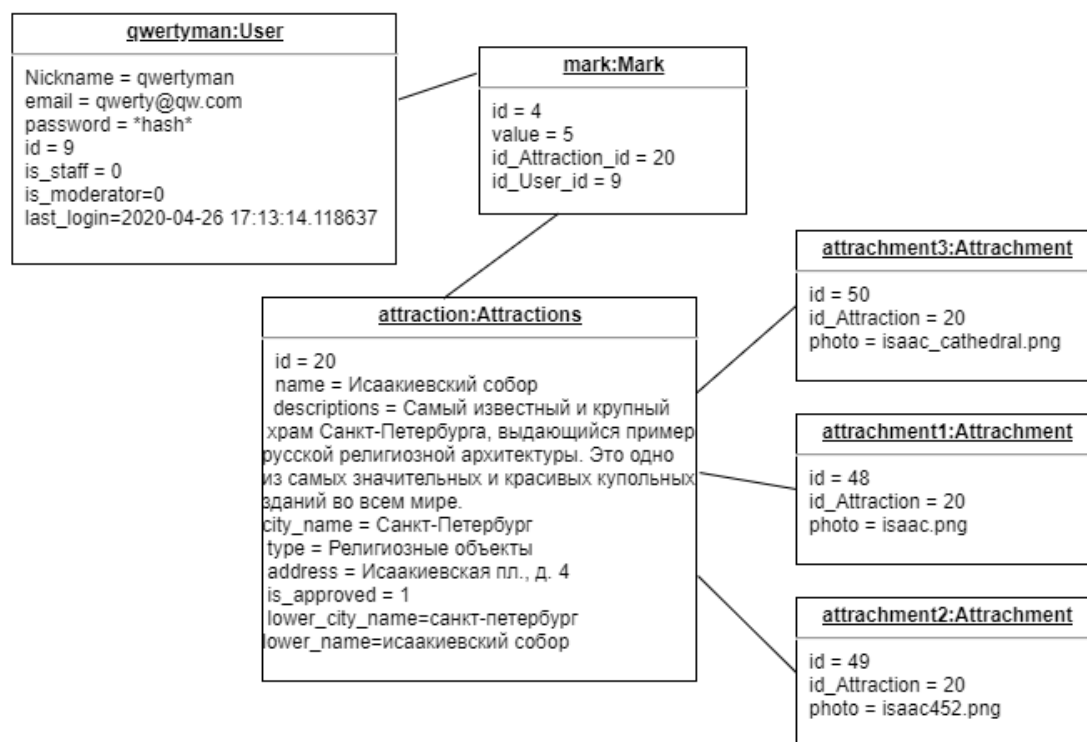


Рисунок 5 - Диаграмма объектов

6.5. Диаграмма последовательностей

Диаграмма последовательностей отображает взаимодействие объектов в динамике.

На первой мы видим взаимодействие Незарегистрированного пользователя при Регистрации. НП передает данные в систему, после этого система делает запрос к БД и получает от нее ответ, содержащий либо информацию о добавлении нового Пользователя, либо о невозможности данного действия в связи с существованием такого Пользователя в БД. При успешном сценарии происходит перенаправление на страницу логина.

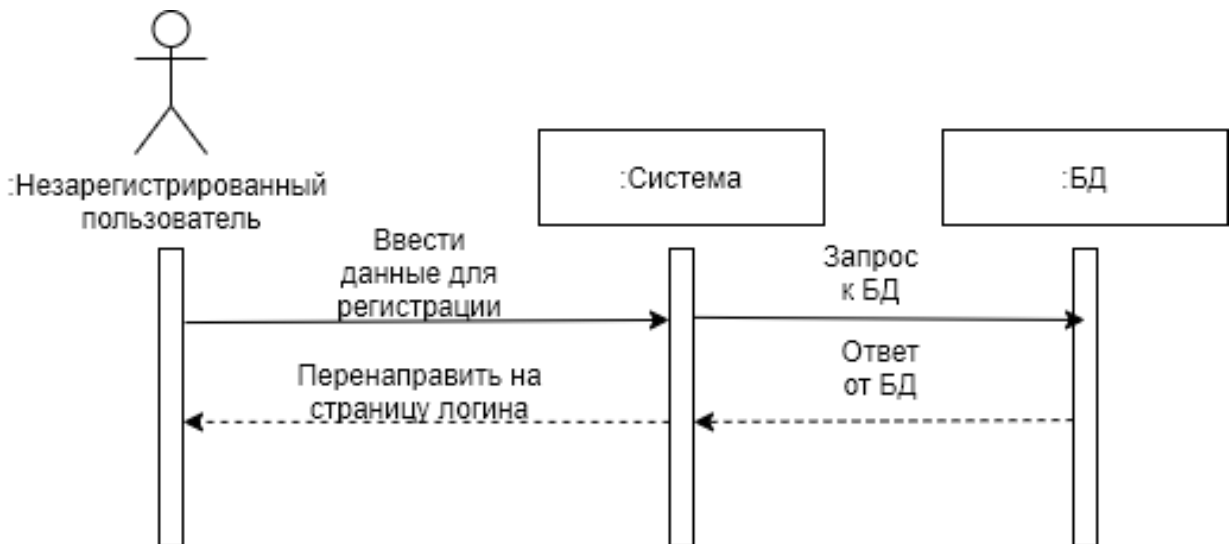


Рисунок 6 - Диаграмма последовательностей для регистрации

На второй диаграмме показано взаимодействие Авторизованного пользователя при выставлении оценки. АП выставляет оценку, путем нажатия на «звездочку», данная оценка передается в Систему. После чего происходит запрос к БД о добавлении этой оценки, система получает ответ от БД, после чего происходит обновление оценки у АП.



Рисунок 7 - Диаграмма последовательностей для выставления оценки

На третьей показано взаимодействие в процессе добавления Достопримечательности. АП отправляет Заявку в Систему, после чего происходит запрос к БД на добавление Заявки. Модератор для просмотра Заявки обращается к Системе, которая делает запрос к БД, Система получает ответ от БД, после чего Заявка отправляется Модератору.

Модератор обращается к Системе, принимая или одобряя Заявку. Система же делает запрос к БД.

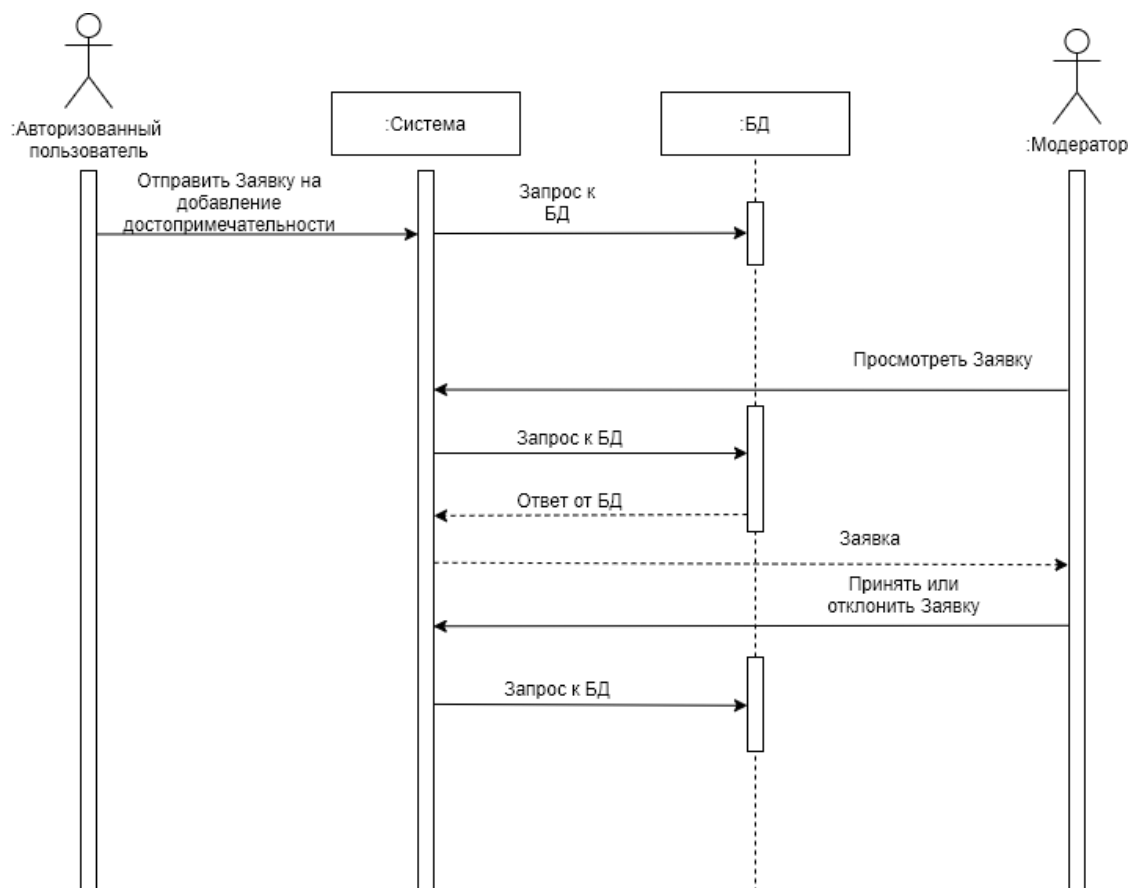


Рисунок 8 - Диаграмма последовательностей для Заявки на добавление

6.6. Диаграмма коопераций

С помощью диаграммы кооперации можно описать полный контекст взаимодействий как своеобразный временной "среза" совокупности объектов, взаимодействующих между собой для выполнения определенной задачи или бизнес-цели программной системы.

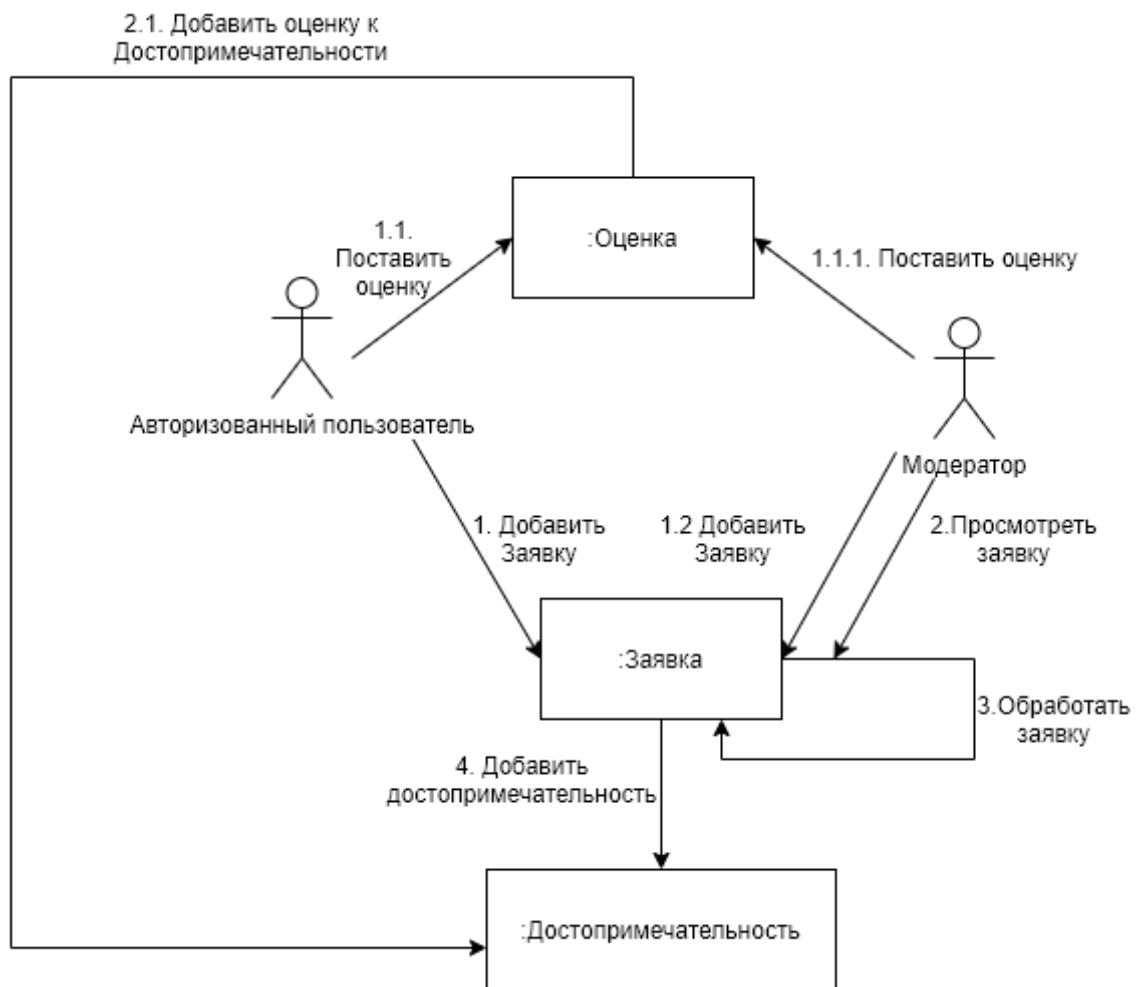


Рисунок 9 - Диаграмма коопераций

6.7. Диаграмма состояний

Они определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате влияния некоторых событий.

На рис. 10 мы видим состояния Заявки на добавление Достопримечательности. На первых этапах она «рассматривается» и «проверяется», после чего, если она прошла проверку, получает статус «Принята» и далее переходит в свое конечное состояние «Доступна всем пользователям».

Иначе, при не прохождении проверки, статус изменяется на «Отклонена» и конечным состоянием будет «Удалена».

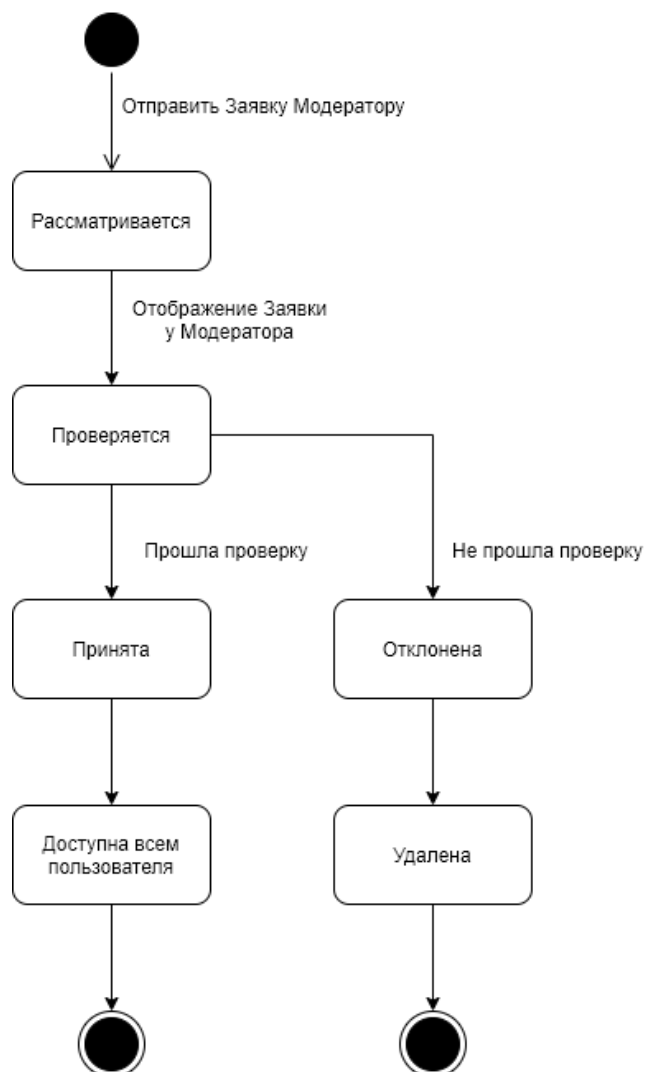


Рисунок 10 - Диаграмма состояний для Заявки

6.8. Диаграмма активностей

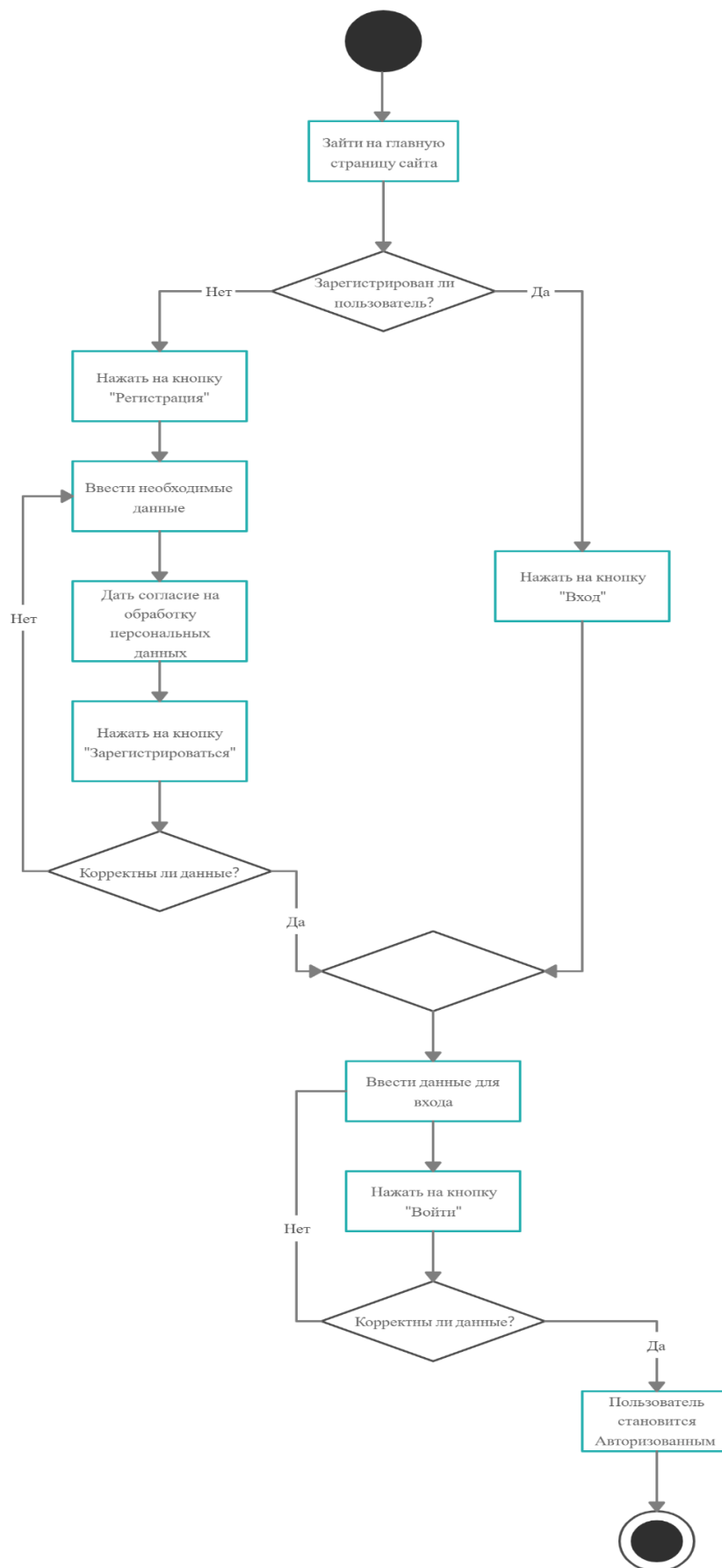


Рисунок 11 - Диаграмма активности для регистрации и авторизации

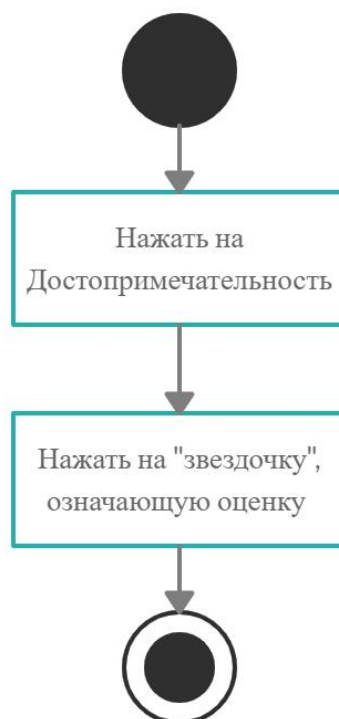


Рисунок 12 - Диаграмма активности для выставления оценки



Рисунок 13 - Диаграмма активности для выхода из профиля

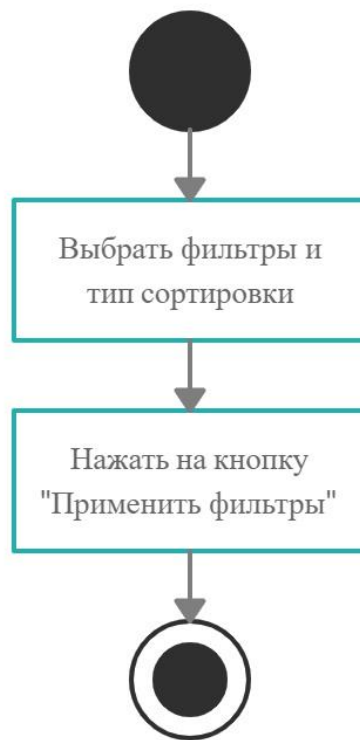


Рисунок 14 - Диаграмма активности для применения фильтров

6.9. Диаграмма развертывания

Диаграмма развертывания предназначена для визуализации элементов и компонентов системы, существующих лишь на этапе ее исполнения, к которым относятся исполнимые файлы, динамические библиотеки, таблицы БД и т. д.

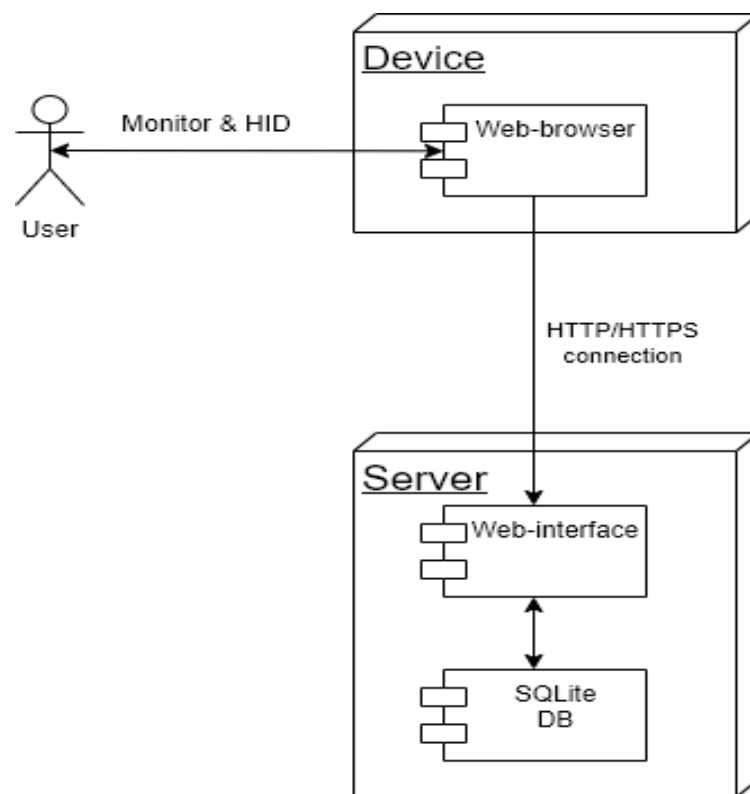


Рисунок 15 - Диаграмма развертывания

7. Анализ ключевых сценариев

Чтобы выяснить, насколько приложение отвечает потребностям пользователей, к нему подключается система веб-аналитики. Получая данные о посетителях, мы можем анализировать их действия и на основе выводов из этих данных принимать решения о доработке системы для увеличения конверсии.

В системе аналитики будет вестись учет трех ключевых вариантов использования:

- Регистрация
- Создание Заявки на добавление Достопримечательности в Справочник
- Выставление Оценки

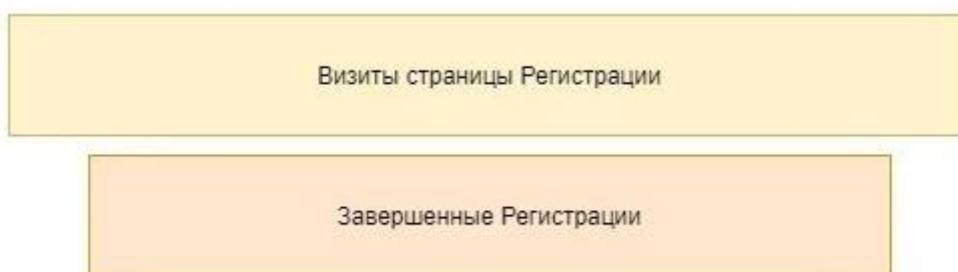


Рисунок 16 - Воронка "Регистрация"

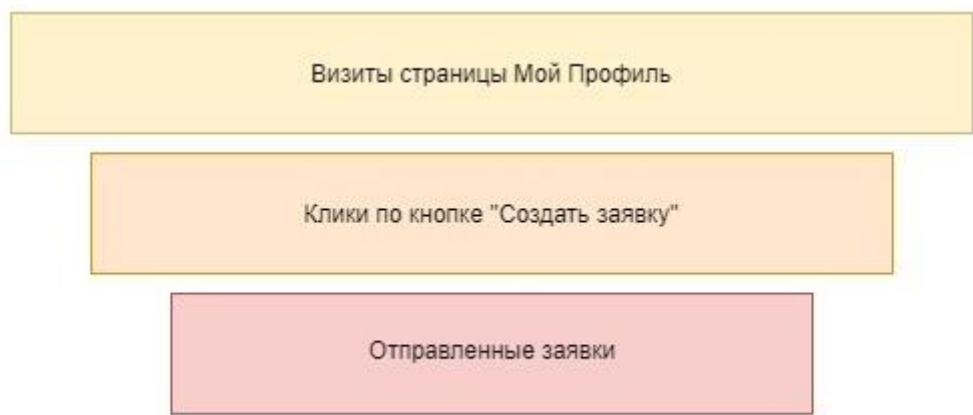


Рисунок 17 - Воронка "Создание заявки на добавление"

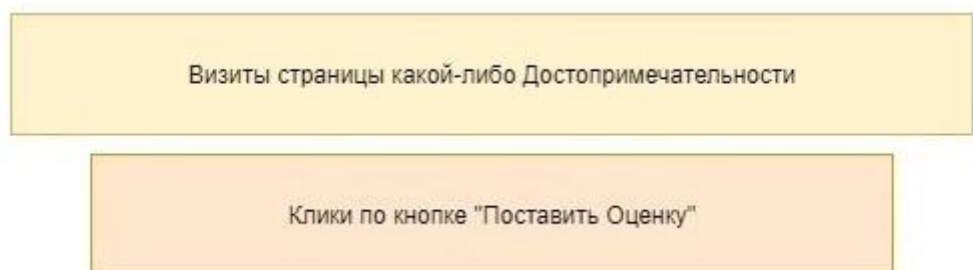


Рисунок 18- Воронка "Выставление оценки"

8. Реализация приложения

8.1. Выбор используемых технологий

Для работы проекта были выбраны следующие используемые технологии:

Во-первых, был выбран язык программирования Python в связи с тем, что данный язык является одним из наиболее популярных в данное время, что значительно повышает вероятность нахождения квалифицированного специалиста для поддержки Справочника. Будет использоваться фреймворк Django, так как в нем реализовано множество типовых блоков, на создание и тестирование которых не придется тратить много времени, что позволит сконцентрироваться на решении поставленной задачи - создании Справочника. Также в Django реализовано объектно-реляционное отображение, что позволяет передавать данные из БД в код приложения. Еще одним преимуществом является разделение бизнес-логики и визуальной части на уровне архитектуры.

Во-вторых, была выбрана СУБД SQLite, так как является встроенным решением в используемый фреймворк Django, исходный код находится в публичном доступе - следовательно нет никаких ограничений на использование данной СУБД, а также существует множество встроенных функций.

В-третьих, одним из используемых решений был выбран Bootstrap. По сути, он представляет с собой фреймворк, содержащий компоненты CSS, HTML и JavaScript, а также собственные стили и шрифты. Данное решение было выбрано в связи с тем, что, во-первых, его использование сокращает время, необходимое для создания макета сайта, так как предлагает достаточное количество шаблонов, во-вторых, адаптивность, то есть сайт

будет идентично отображаться на разных устройствах и в различных браузерах.

В-четвертых, системой аналитики, как указывалось выше, подключаемой к нашему Справочнику, была выбрана Яндекс.Метрика. Данный сервис позволяет осуществлять анализ аудитории веб-ресурса, конверсии сайта, а также доступности сайта и скорости его работы. Был выбран этот сервис в связи с тем, что Яндекс.Метрика позволяет не только получить общее представление о посещаемости того или иного ресурса, но и даёт возможность получения детальных данных. Так, Метрика может осуществлять группировку посетителей по тем или иным признакам и параметрам. Также можно узнать о том, какие интернет-страницы сайта являются наиболее востребованными среди посетителей.

В-пятых, по требованию заказчика к системе был добавлен Swagger.

8.2. Интерфейс

При заходе на сайт, незарегистрированный пользователь, также и неавторизованный попадает на главную страницу сайта.

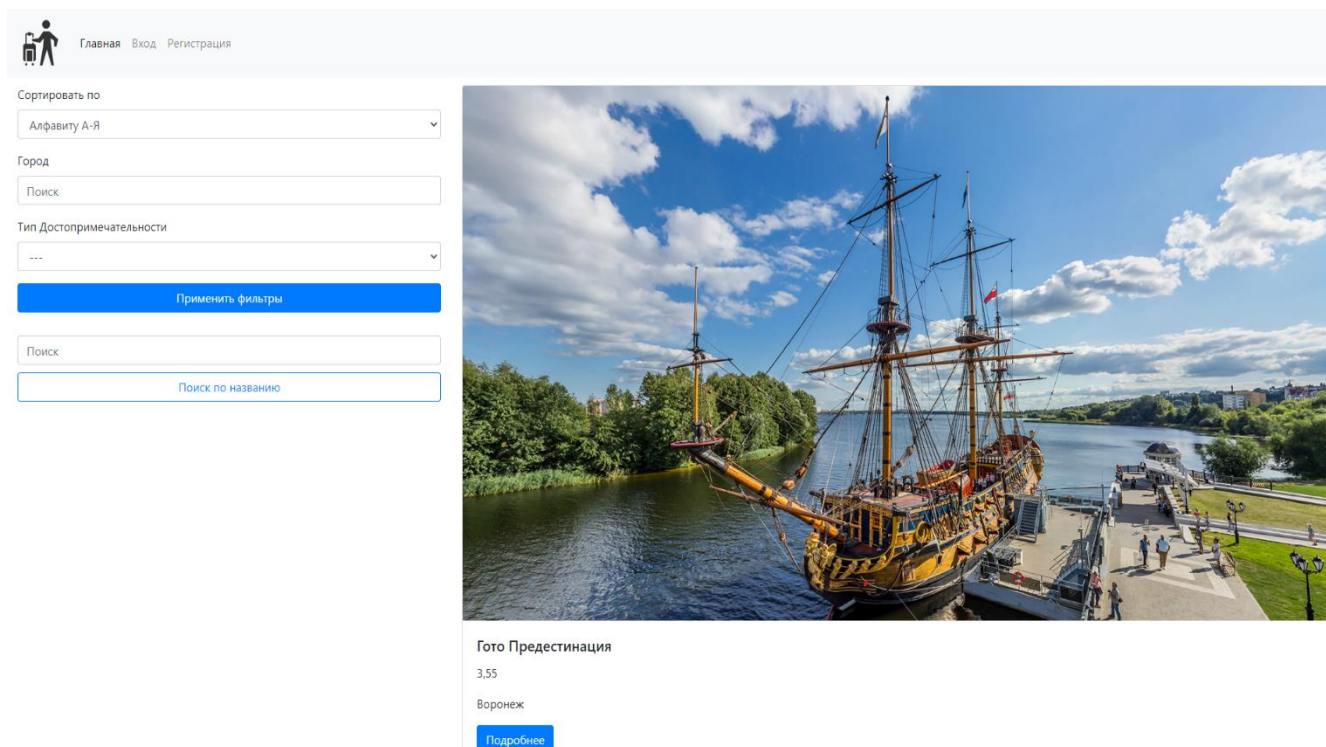


Рисунок 19 – Главная страница НП

На ней он может ознакомиться со списком достопримечательностей, применить фильтры, осуществить поиск по названию города или названию самой Достопримечательности. Также, он может открыть страницу Достопримечательности и ознакомиться с основной информацией о ней, а также узнать среднюю оценку.

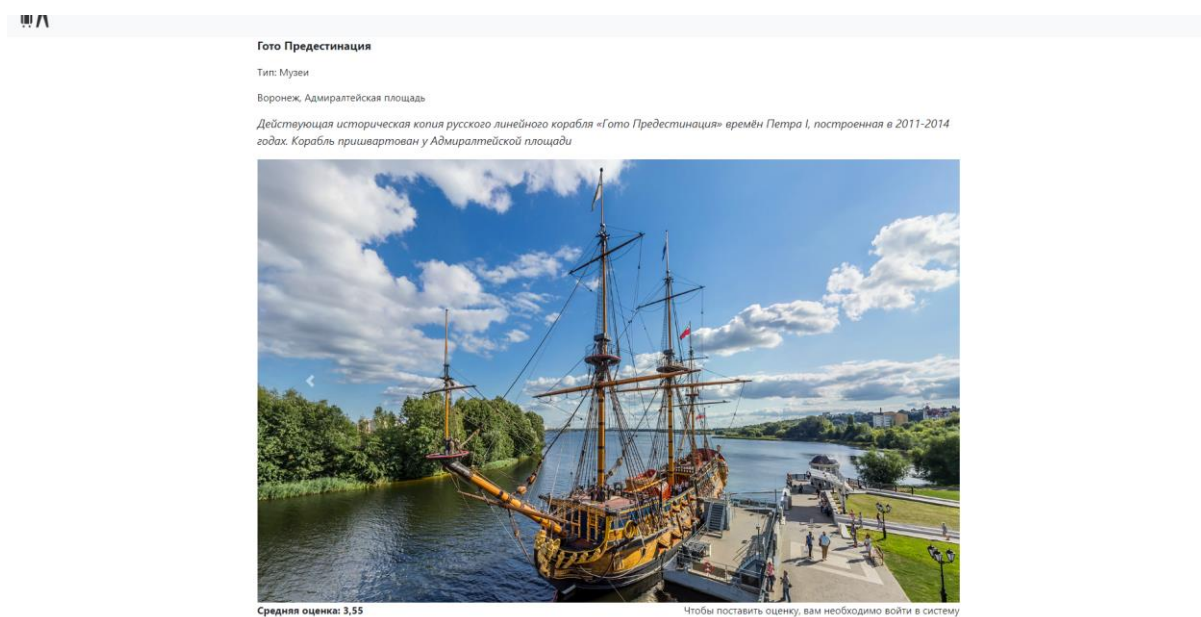
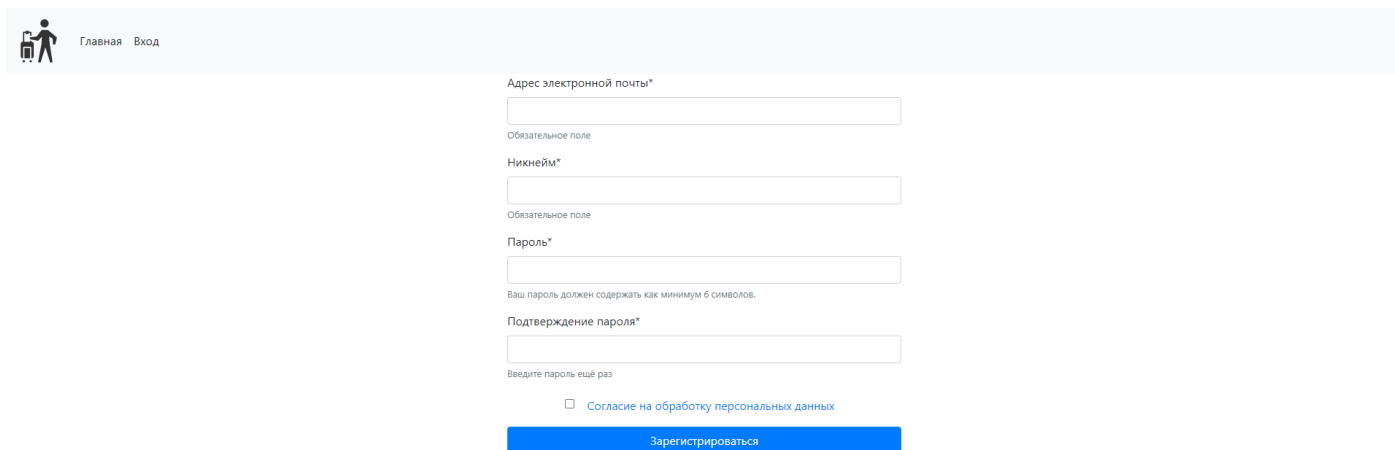


Рисунок 20 – Страница Достопримечательности НП

Как мы видим, чтобы поставить оценку необходимо авторизоваться. Для этого, если пользователь еще не зарегистрирован, ему необходимо нажать на кнопку «Зарегистрироваться» и ввести необходимые данные, а также подтвердить согласие на обработку персональных данных.



Главная Вход

Адрес электронной почты*

Обязательное поле

Никнейм*

Обязательное поле

Пароль*

Ваш пароль должен содержать как минимум 6 символов.

Подтверждение пароля*

Введите пароль ещё раз

☐ [Согласие на обработку персональных данных](#)

[Зарегистрироваться](#)

Рисунок 21 – Страница Регистрации

После успешной регистрации пользователь попадает на страницу входа. Также важно отметить, что пользователь может перейти на данную страницу без посещения страницы регистрации, если его данные уже существуют в системе и ему необходимо лишь авторизоваться.

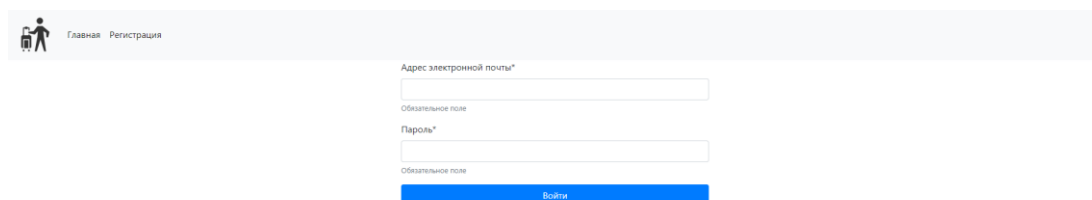


Рисунок 22 – Страница Входа

После этого он становится АП и попадает на свою главную страницу, которая видоизменена относительно главной, рассмотренной выше.

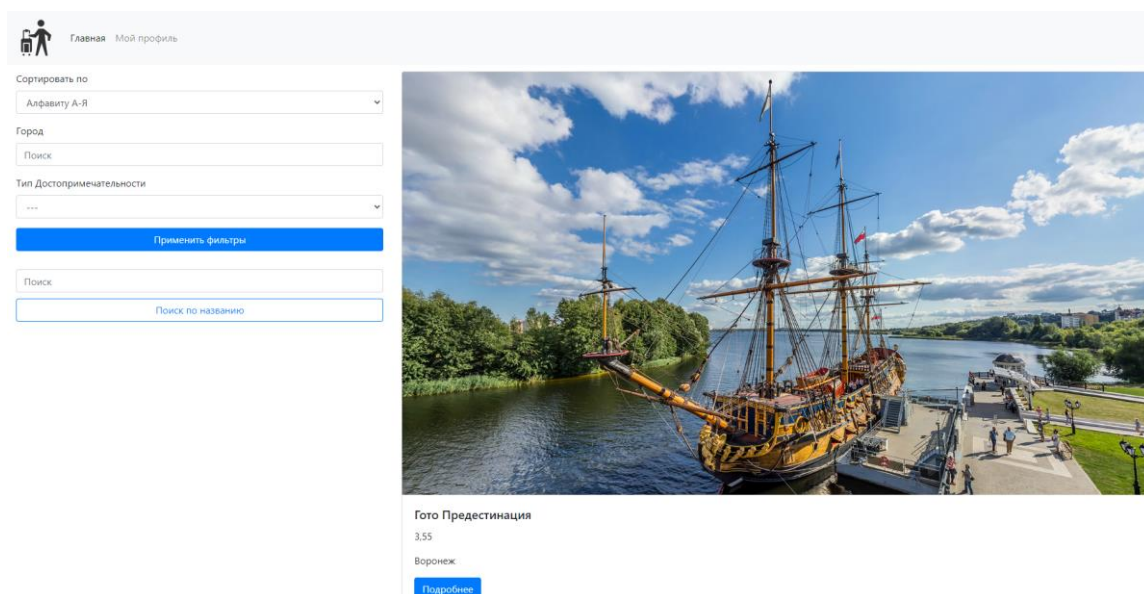


Рисунок 23 – Главная страница АП

Кроме действий, доступных для неавторизованных и незарегистрированных пользователей, он также может перейти на страницу своего профиля, а также самостоятельно оценить Достопримечательности Справочника.

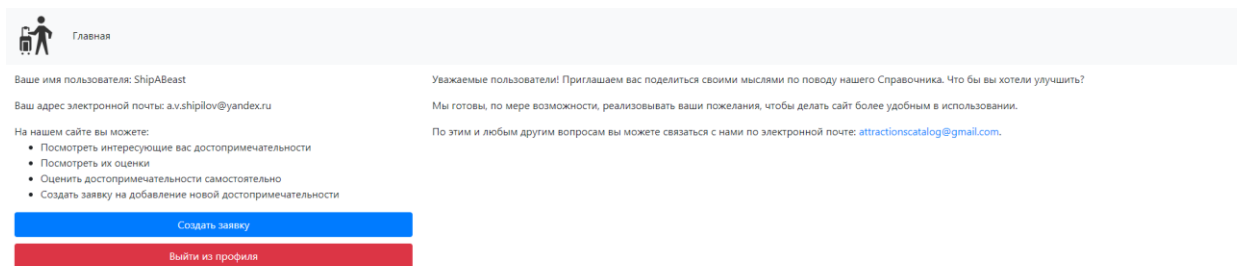


Рисунок 24 – Страница «Мой профиль»

С данной странице он может перейти на страницу создания заявки, которая будет отправлена модератору после ввода данных.

Правила Оформления Заявок

1. Следует создавать заявку лишь для достопримечательности, которая отсутствует в Справочнике.
2. Введенные данные должны быть верны, а именно: название, адрес, описание, город, тип, фото. Они должны быть согласованы между собой и соответствовать действительности, то есть соответствовать реально существующим достопримечательностям.
3. Описание должно содержать краткую информацию о достопримечательности и составлять не более 300 символов.
4. Поля должны быть заполнены без орфографических ошибок.
5. Необходимо добавить 1-3 фотографии достопримечательности, с разрешением не ниже 1280x720 пикселей и не более 1920x1080. Размер файлов - не более 5Мб.
6. Запрещено использовать ненормативную, запрещенную, оскорбительную и разжигающую ненависть лексику.
7. Запрещена любая реклама и продакт-плейсмент.
8. Публикуйте фото, уместные для многогранной аудитории. В число запрещенных входят фото, изображающие половые акты, половые органы и полностью обнаженных людей. Изображения обнаженных тел в живописи и скульптуре разрешены.

При несоблюдении любого из пунктов, заявка будет отклонена.

Рисунок 25 – Страница «Создать заявку»

Страница Достопримечательностей для АП имеет следующий вид:

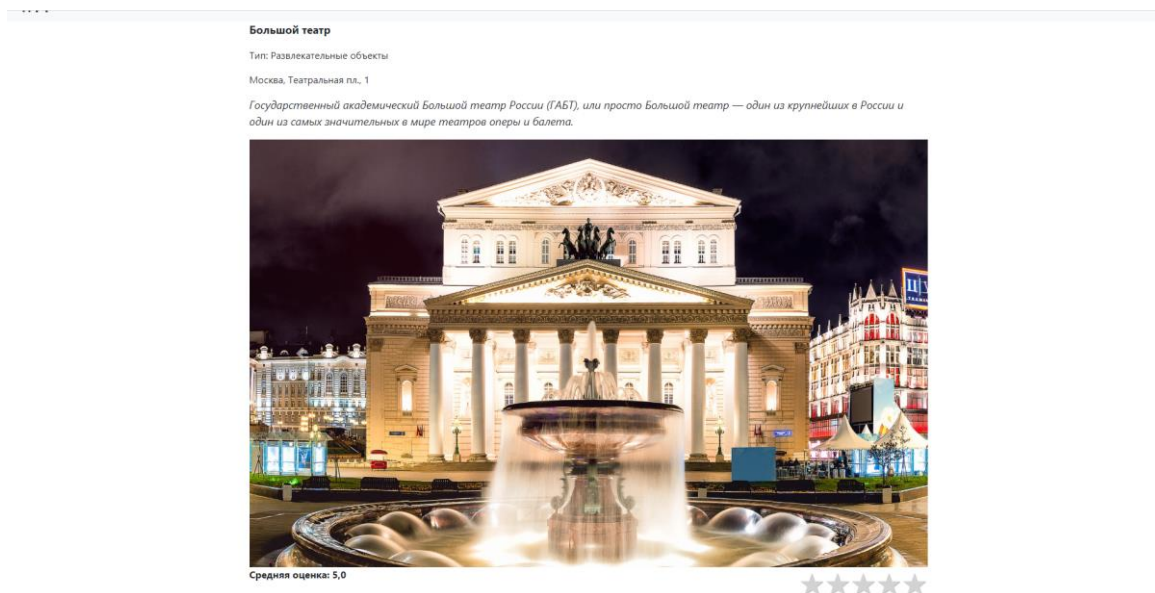


Рисунок 26 – Страница Достопримечательности АП

Страницы модерации заявок имеют следующий вид:

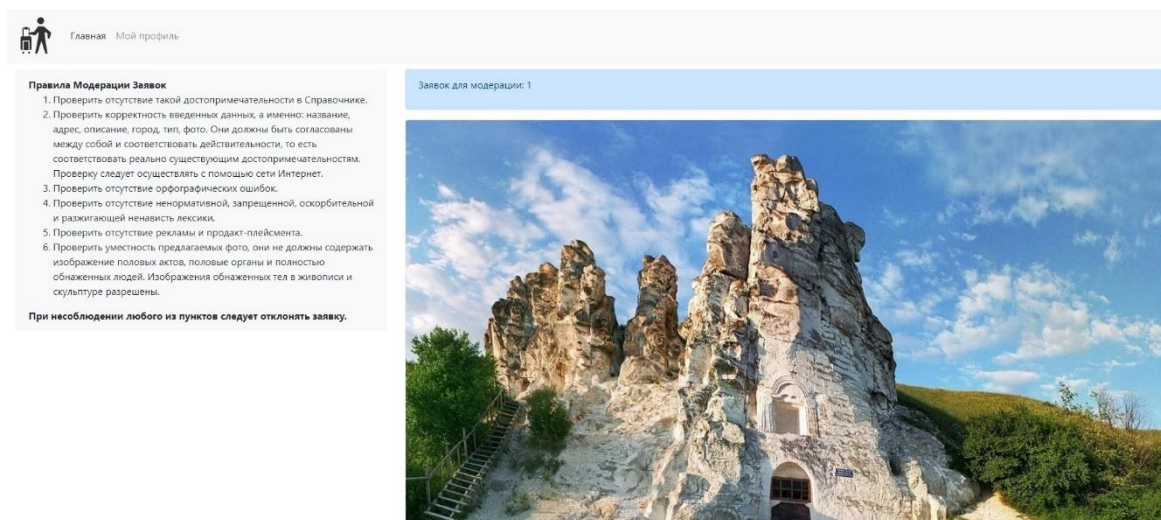


Рисунок 27 – Страница предложенных Достопримечательностей

Дивногорье

Тип: Архитектура и памятники

Воронежская область, Воронежская область, Лискинский район, хутор Дивногорье

Пещерный комплекс в Больших Дивах расположен в основании мелового останца-дивы. Комплекс имеет два яруса, объединенных лестничным переходом.

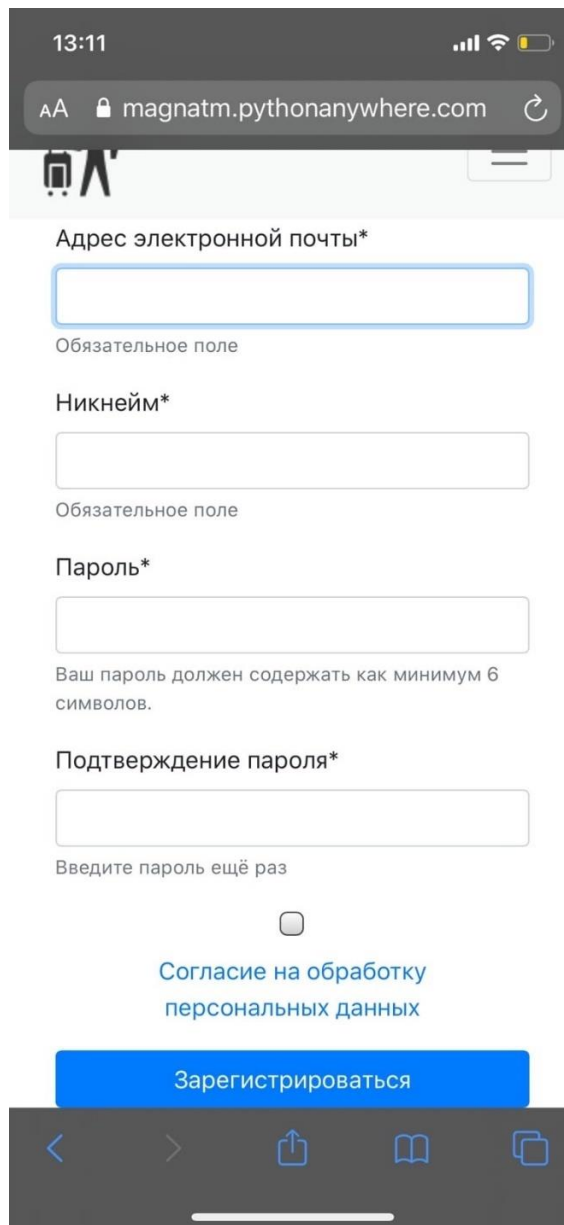


Отклонить

Принять

Рисунок 28 – Страница Модерации

Наше приложение оптимизировано под мобильные платформы и планшетные компьютеры. Ниже представлены некоторые страницы, открытые на iPhone X с помощью браузера Safari.



13:11

AA magnatm.pythonanywhere.com

Адрес электронной почты*

Обязательное поле

Никнейм*

Обязательное поле

Пароль*

Ваш пароль должен содержать как минимум 6 символов.

Подтверждение пароля*

Введите пароль ещё раз

☐

[Согласие на обработку персональных данных](#)

[Зарегистрироваться](#)




< >   

Рисунок 29 – Страница Регистрации на мобильных платформах

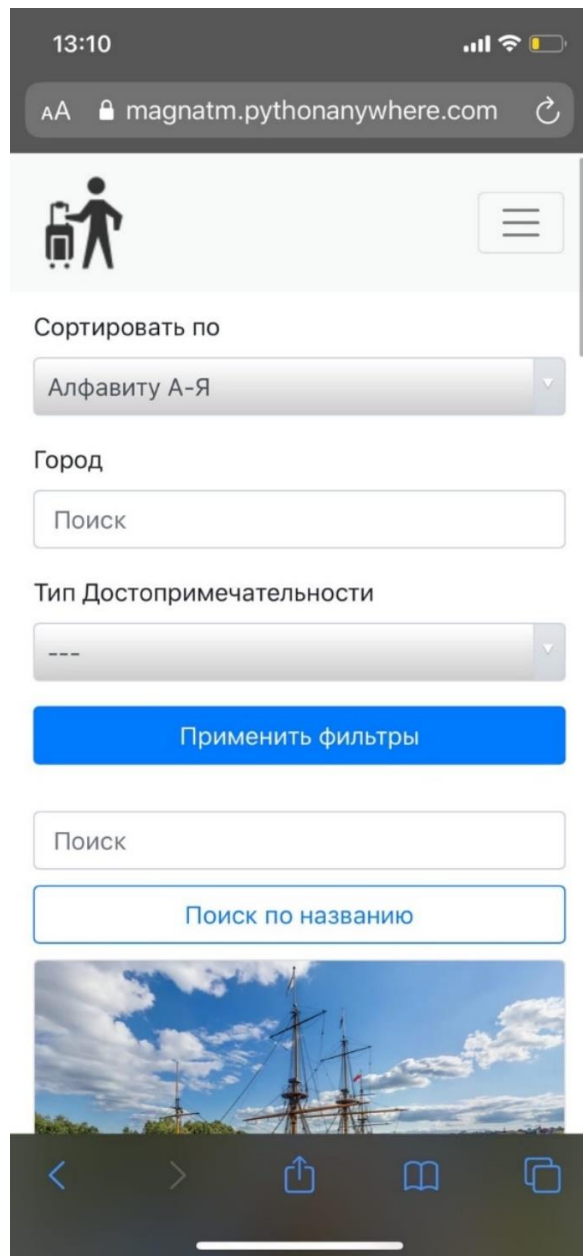


Рисунок 30 – Главная страница на мобильных платформах

8.3. Система аналитики

Как указывалось выше, к нашему проекту подключена система аналитики Яндекс.Метрика.

В данной системе ведется учет всех 3 ключевых воронок, а именно: регистрация, создание заявки на добавление и выставление оценки. Также, ведется сбор и другой информации.

Ниже представлены рисунки, иллюстрирующие работу Метрики.

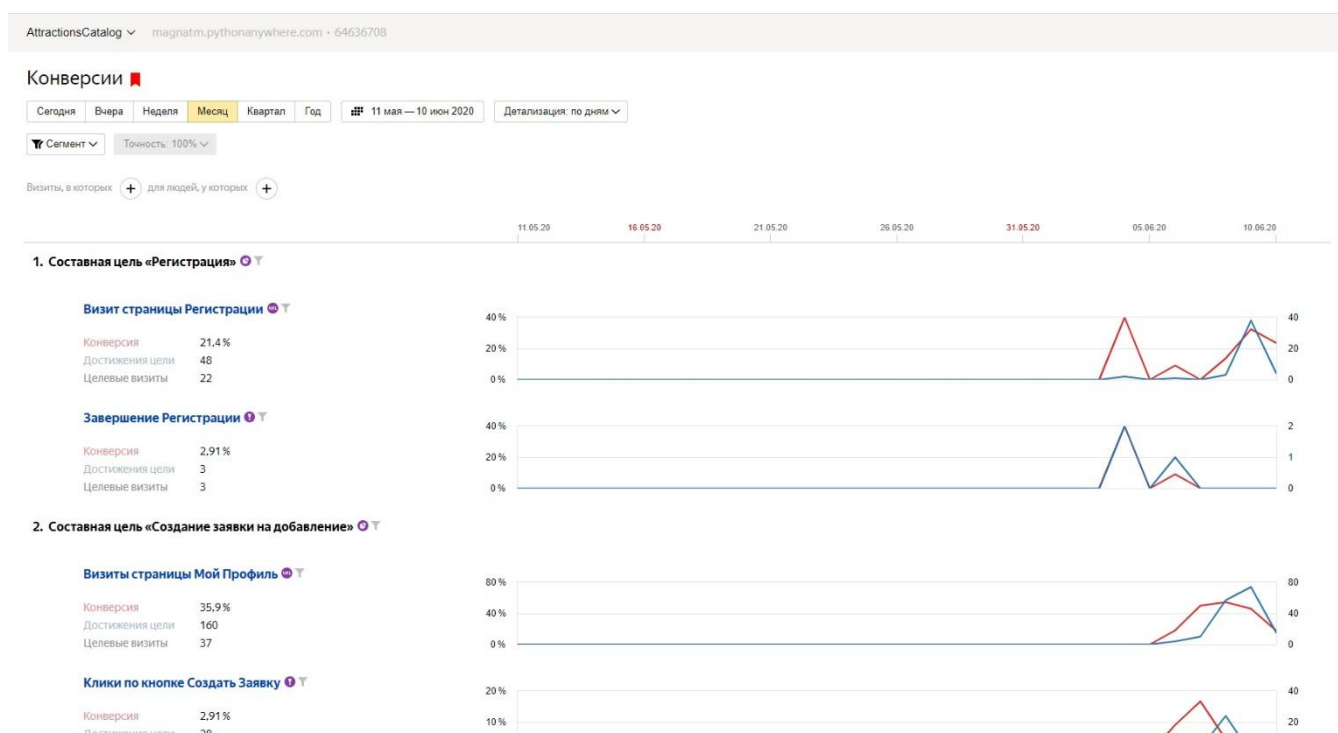


Рисунок 31 – Конверсии

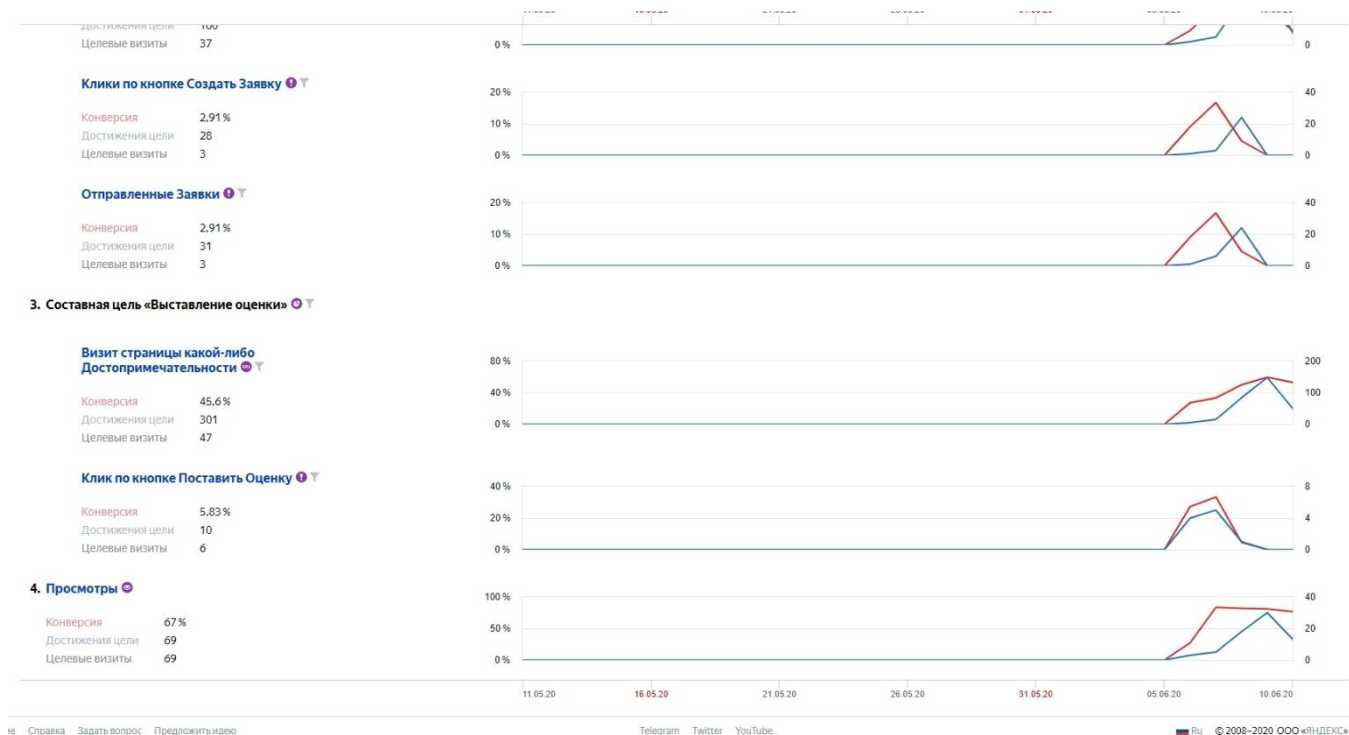


Рисунок 32 – Конверсии

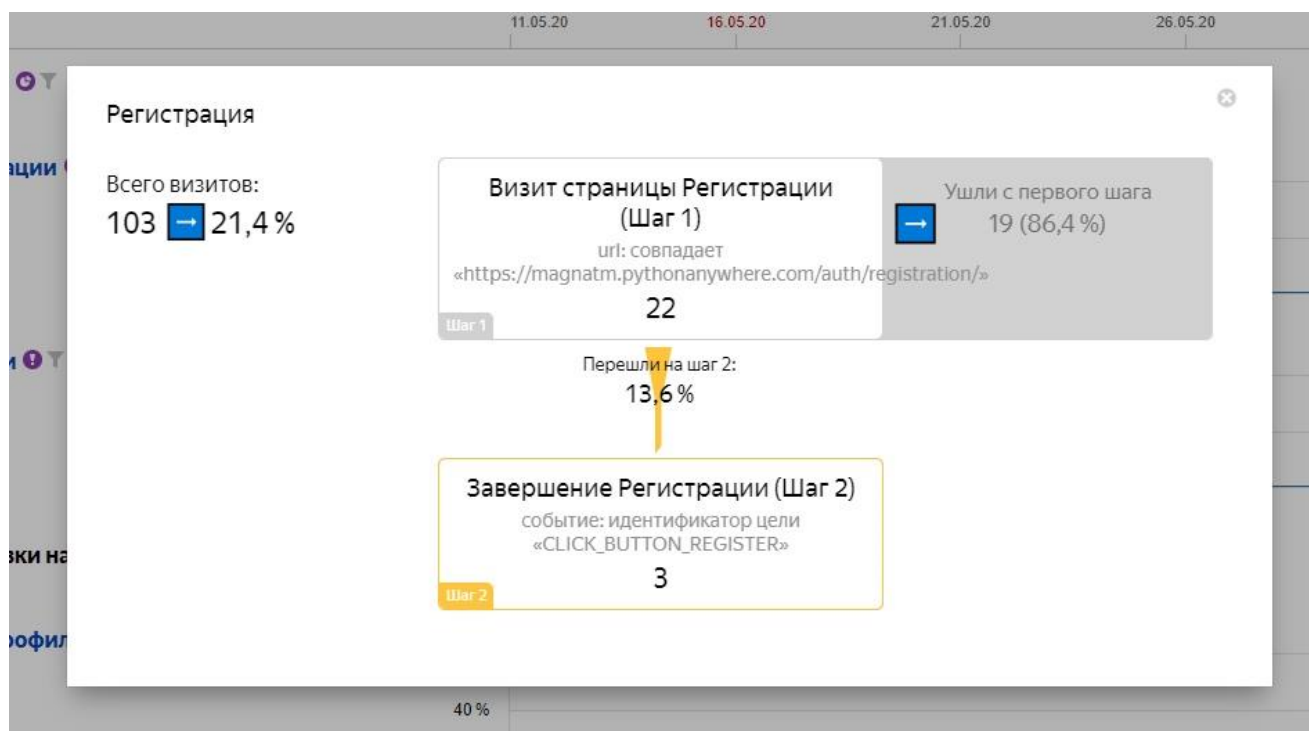


Рисунок 33 – Воронка «Регистрация» в Метрике

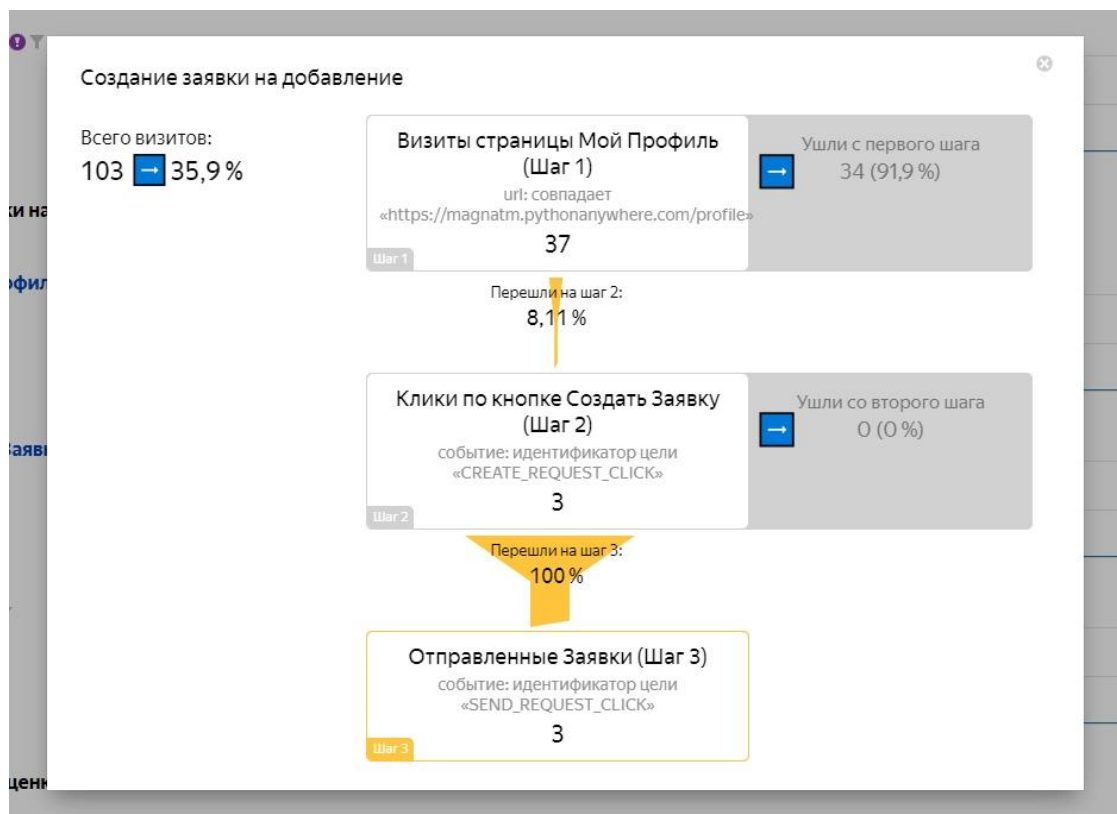


Рисунок 34 – Воронка «Создание заявки на добавление» в Метрике

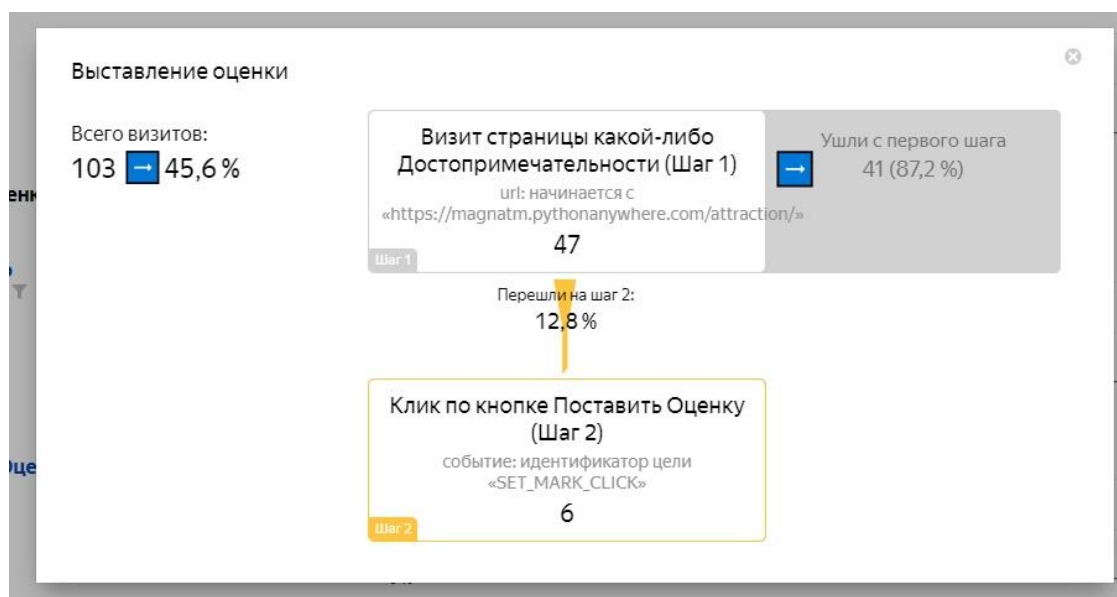


Рисунок 35 – Воронка «Выставление оценки» в Метрике

9. Тестирование

Как и было заявлено, наша система подверглась тестированию, отчет представлен ниже.

Было произведено модульное тестирование с использованием фреймворка unittest, результаты предоставлены ниже.

ID	Название	Статус
TC_AttrView_01	Доступ к странице существующей достопримечательности	Passed
TC_AttrView_02	Доступ к странице несуществующей достопримечательности	Passed
TC_AttrView_03	Доступ к странице логина	Passed
TC_AttrView_04	Доступ к странице регистрации	Passed
TC_AttrView_05	Доступ к странице "Мой профиль" (без аутентификации)	Passed
TC_AttrView_06	Доступ к странице "Мой профиль" (с аутентификацией)	Passed
TC_AttrView_07	Регистрация	Passed with minors
TC_AttrLogin_01	Логин - верные данные	Passed
TC_AttrLogin_02	Логин - неверный емейл	Passed
TC_AttrLogin_03	Логин - неверный пароль	Passed
TC_FormLogin_01	Отправка валидной формы	Passed
TC_FormLogin_02	Отправка формы без обязательного атрибута	Passed
TC_FormLogin_03	Поле email- ограничения	Passed
TC_FormLogin_04	Поле email- ограничения	Passed
TC_FormLogin_05	Поле password- ограничения	Passed
TC_FormLogin_06	Поле password- ограничения	Passed
TC_FormRegister_01	Отправка валидной формы	Passed
TC_FormRegister_02	Отправка формы с несовпадающими паролями	Passed
TC_FormRegister_03	Отправка формы с пустым полем	Passed
TC_FormRegister_04	Отправка формы с пустым полем	Passed
TC_FormRegister_05	Отправка формы с коротким паролем	Passed
TC_FormRegister_06	Проверка ограничений длины(негативная)	Passed
TC_FormRegister_07	Проверка ограничений длины	Passed
TC_FormCreateAttraction_01	Проверка валидной формы	Passed

TC_FormCreateAttraction_02	Проверка невалидной формы	Passed
TC_FormCreateAttraction_03	Проверка ограничений длины	Passed
TC_FormCreateAttraction_04	Проверка ограничений длины (негативная)	Passed

Таблица 2 – Результаты тестирования

При помощи инструмента coverage был получен процент покрытия кода нашего приложения модульными тестами.

На рисунке ниже можно увидеть полученное тестовое покрытие нашего приложения.

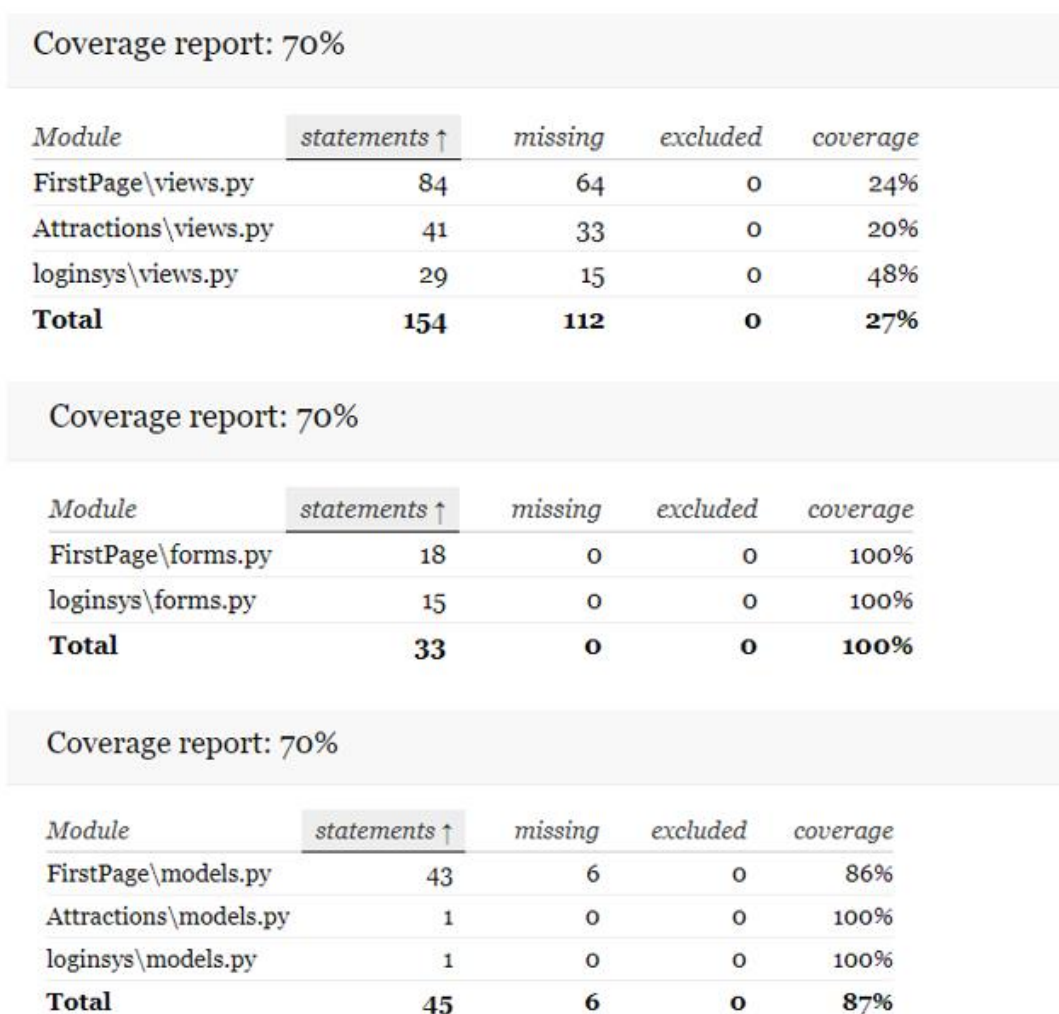


Рисунок 36 – Тестовое покрытие

Тест кейсы для модульного тестирования можно просмотреть в Приложении А.

Ручное тестирование приложения на уязвимости типа XSS и SQL-инъекций производилось на случаях, представленных ниже:

Все значения из списка ниже были вписаны в поля ввода (поле для регистрации, входа, поля поиска и фильтрации, а также поля формы оформления заявки):

```
1.      <?xml version="1.0" encoding="ISO-8859-1"?><!DOCTYPE
foo[<!ELEMENT foo ANY><!ENTITY xxe SYSTEM
"file:///etc/passwd">]><foo>&xxe;</foo>
2.      `"(
3.      sleep(7)#1
4.      ;env;
5.      `; DROP TABLE FirstPage_mark; --
6.      " OR 1=1;--
7.      
8.      <script>alert(xss)</script>
9.      User' or 1=1);--
10.     1' AND IF(MID(VERSION(),1,1)='5', sleep(15),0) --
```

Не было найдено уязвимостей при проверке значений из данного списка.

Для более полного покрытия, тестирование сайта на уязвимости было произведено при помощи утилиты Wapiti.

Данная утилита в автоматическом режиме тестирует сайт на уязвимости (в том числе SQL-инъекции и XSS-атаки)

Отчет работы данной утилиты можно увидеть на рисунке ниже:

Wapiti vulnerability report

Target: <https://magnatm.pythonanywhere.com/>

Date of the scan: Wed, 10 Jun 2020 09:14:24 +0000. Scope of the scan: folder

Summary

Category	Number of vulnerabilities found
SQL Injection	0
Blind SQL Injection	0
File Handling	0
Cross Site Scripting	0
CRLF Injection	0
Commands execution	0
Htaccess Bypass	0
Backup file	0
Potentially dangerous file	0
Server Side Request Forgery	0
Open Redirect	0
XXE	0

Рисунок 37 – Отчет работы утилиты Wapiti

Также было произведено тестирование удобства использования. Для этого была выбрана группа из пяти человек. Им было предложено воспользоваться сайтом и ответить на вопросы об использовании сайта.

Также был пункт пожеланий и предложений. Пример тестового варианта представлен в Приложении Б.

Оценка производилась по 5 балльной шкале в целых величинах.

Средняя оценка по отзывам пользователей получилась 3.9

После проведения тестирования было решено изменить вариант использования “Модерация заявок”. 5 из 5 человек предложили перенести критерии приемки заявки на страницу со списком всех заявок для модерации.

Таким образом были проведены работы по тестированию программы, соответствующие заявленным ранее в техническом задании.

10. Заключение

Таким образом, в ходе данного проекта были выполнены основные цели, а именно, создан реестр достопримечательностей России, упрощающий получение информации о них Пользователями, а значит, ведущий к повышению уровня заинтересованности.

Наш Справочник, созданный в соответствии с Техническим Заданием, утвержденным заказчиком на первом этапе работы, соответствует всем заявленным основным требованиям, а именно: предоставляет Пользователю возможность просмотра Достопримечательностей, их фильтрации, сортировки и поиска, также предоставляет АП возможность отправки заявки на добавление Достопримечательности, оценки Достопримечательностей.

11. Перечень принятых определений и сокращений

Достопримечательность - место, вещь или объект, заслуживающие особого внимания, знаменитые или замечательные чем-либо, например, являющиеся историческим наследием, художественной ценностью. В системе представлена как запись в БД, содержащая название, описание, адрес, название города, тип достопримечательности.

Справочник Достопримечательностей (или Справочник, Система) - хранилище Достопримечательностей.

ТЗ - техническое задание.

Пользователь (или Незарегистрированный Пользователь) - лицо, использующее Справочник и не прошедшее авторизацию.

Авторизованный пользователь (или АП) - Пользователь, который вошел в систему, используя Логин и Пароль.

Регистрация - внесение данных Пользователя в БД.

База данных (или БД) - организованная структура, предназначенная для хранения, изменения и обработки информации.

Неавторизованный Пользователь (или НП) - Пользователь, прошедший регистрацию, но не авторизовавшийся в систему.

Логин - электронная почта(email).

Пароль - набор символов, предназначенный для входа в Систему.

Модератор - АП, который способен обрабатывать заявки и добавлять достопримечательности.

Заявка - сформированное АП обращение на добавление новой Достопримечательности.

Профиль - Страница с информацией о пользователе.

Оценка - мнение Пользователя о достопримечательности, выраженное по 5-балльной шкале(1-низшая оценка, 5- наивысшая).

Конверсия — это отношение числа посетителей сайта, выполнивших на нём какие-либо целевые действия, к общему числу посетителей сайта, выраженное в процентах.

Приложение А

#	Шаги	Ожидаемый результат	Пояснения
TC_AttrView_01	Доступ к странице существующей достопримечательности		
	1 Создать тестового клиента		Тестовый клиент
	2 Указать URL существующей достопримечательности		/attraction/{id}
	3 Запрос на данную страницу		resp = self.client.get(url)
	4 Получить ответ о переходе на данную страницу	Код ответа - 200	"Успешно". Запрос успешно обработан
TC_AttrView_02	Доступ к странице несуществующей достопримечательности		
	1 Создать тестового клиента		Тестовый клиент
	2 Указать URL несуществующей достопримечательности		/attraction/{id}
	3 Запрос на данную страницу		resp = self.client.get(url)
	4 Получить ответ о переходе на данную страницу	Код ответа - 404	Страница не найдена
TC_AttrView_03	Доступ к странице логина		
	1 Создать тестового клиента		Тестовый клиент
	2 Указать URL страницы логина		/auth/login/
	3 Запрос на данную страницу		resp = self.client.get(url)
	4 Получить ответ о переходе на данную страницу	Код ответа - 200	"Успешно". Запрос успешно обработан
TC_AttrView_04	Доступ к странице регистрации		
	1 Создать тестового клиента		Тестовый клиент
	2 Указать URL страницы регистрации		/auth/registration/
	3 Запрос на данную страницу		resp = self.client.get(url)
	4 Получить ответ о переходе на данную страницу	Код ответа - 200	"Успешно". Запрос успешно обработан
TC_AttrView_05	Доступ к странице "Мой профиль" (без аутентификации)		
	1 Создать тестового клиента		Тестовый клиент
	2 Указать URL страницы профиля		/profile/
	3 Запрос на данную страницу		resp = self.client.get(url)
	4 Получить ответ о переходе на данную страницу	Код ответа - 403	"Запрещено". У клиента нет прав доступа к содержимому
TC_AttrView_06	Доступ к странице "Мой профиль" (с аутентификацией)		
	1 Создать тестового клиента		Тестовый клиент
	2 Войти в систему	Аутентификация прошла успешно	Использовать "self.client.force_login"
	3 Указать URL страницы профиля		/profile/
	4 Запрос на данную страницу		resp = self.client.get(url)
	5 Получить ответ о переходе на данную страницу	Код ответа - 200	Успешно. Запрос успешно обработан

Рисунок А.1 – Тест кейсы

	5 Получить ответ о переходе на данную страницу	Код ответа - 200	Успешно. Запрос успешно обработан
TC_AttrView_08	Регистрация		
	1 Создать тестового клиента		Тестовый клиент
	2 post запрос на страницу регистрации		self.client.post("/auth/registration/", { 'email': 'new@mnnn.com', 'nickname': 'nickname', 'password1': 'password', 'password2': 'password', 'personal_data_agreement': 'True'})
	3 Получить ответ о перенаправлении на страницу логина	Код ответа - 302	

Рисунок А.2 – Тест кейсы

#	Шаги	Ожидаемый результат	Пояснения
TC_AttrLogin_01	Логин - верные данные		
	1 Создать Пользователя		Создать пользователя
	2 Аутентификация, используя email и пароль из шага 1	Пользователь создан и аутентификация прошла успешно	
TC_AttrLogin_02	Логин - неверный email		
	1 Создать Пользователя		Создать пользователя
	2 Аутентификация, используя пароль из шага 1 и любой другой email	Пользователь создан и аутентификация провалена	
TC_AttrLogin_03	Логин - неверный пароль		
	1 Создать Пользователя		Создать пользователя
	2 Аутентификация, используя email из шага 1 и любой другой пароль	Пользователь создан и аутентификация провалена	

Рисунок А.3 – Тест кейсы

#	Шаги	Ожидаемый результат	Пояснения
LoginForm			
TC_FormLogin_01	Отправка валидной формы		
	1 Сформировать данные для формы: email + password		Данные для формы
	2 Отправить данные в форму	Форма валидна	
TC_FormLogin_02	Отправка формы без обязательного атрибута		
	1 Сформировать данные для формы Отправить только один атрибут		Данные для формы
	2 Отправить данные в форму	Форма невалидна	
TC_FormLogin_03	Поле email-ограничения		
	1 Сформировать данные для формы кол-во знаков в поле email = 50		Данные для формы
	2 Отправить данные в форму	Форма валидна	
TC_FormLogin_04	Поле email-ограничения		
	1 Сформировать данные для формы кол-во знаков в поле email = 51		Данные для формы
	2 Отправить данные в форму	Форма невалидна	
TC_FormLogin_05	Поле password-ограничения		
	1 Сформировать данные для формы кол-во знаков в поле password = 50		Данные для формы
	2 Отправить данные в форму	Форма валидна	
TC_FormLogin_06	Поле password-ограничения		
	1 Сформировать данные для формы кол-во знаков в поле password = 51		Данные для формы
	2 Отправить данные в форму	Форма невалидна	
RegisterForm			
TC_FormRegister_01	Отправка валидной формы		
	1 Сформировать валидные данные для формы: email + nickname + password1+ password2		Данные для формы
	2 Отправить данные в форму	Форма валидна	
TC_FormRegister_02	Отправка формы с несовпадающими паролями		
	1 Сформировать валидные данные для формы: password1 != password2		Данные для формы
	2 Отправить данные в форму	Форма невалидна	
TC_FormRegister_03	Отправка формы с пустым полем		
	1 Сформировать валидные данные для формы: Оставить пустым одно из полей (прим. Проверить для всех полей)		Данные для формы
	2 Отправить данные в форму	Форма невалидна	

Рисунок А.4 – Тест кейсы

TC_FormRegister_04		Отправка формы с пустым полем		
		Сформировать валидные данные для формы: Оставить пустым одно из полей (прим. Проверить для всех полей)		Данные для формы
	1			
	2	Отправить данные в форму	Форма невалидна	
TC_FormRegister_05		Отправка формы с коротким паролем		
		Сформировать валидные данные для формы:		Данные для формы
	1	Пароль < 6 символов		
	2	Отправить данные в форму	Форма невалидна	
TC_FormRegister_06		Отправка формы с коротким паролем		
		Сформировать валидные данные для формы:		Данные для формы
	1	Пароль < 6 символов		
	2	Отправить данные в форму	Форма невалидна	
TC_FormRegister_07		Проверка ограничений длины		
		Сформировать валидные данные для формы: Проверить для пароля, никнейма и email на граничных значениях		Данные для формы
	1			
	2	Отправить данные в форму	Форма валидна	
TC_FormRegister_08		Проверка ограничений длины		
		Сформировать валидные данные для формы: Проверить для пароля, никнейма и email на граничных значениях +1		Данные для формы
	1			
	2	Отправить данные в форму	Форма невалидна	
CreateAttractionForm				
TC_FormCreateAttraction_01		Проверка валидной формы		
		Сформировать валидные данные для формы:		Данные для формы
	1			
	2	Отправить данные в форму	Форма валидна	
TC_FormCreateAttraction_02		Проверка невалидной формы		
		Сформировать невалидные данные для формы: одно из полей оставить пустым		Данные для формы
	1			
	2	Отправить данные в форму	Форма невалидна	
TC_FormCreateAttraction_03		Проверка ограничений длины		
		Сформировать данные для формы: проверить для всех полей на граничных значениях		Данные для формы
	1			
	2	Отправить данные в форму	Форма валидна	
TC_FormCreateAttraction_04		Проверка ограничений длины		
		Сформировать данные для формы: проверить для всех полей на граничных значениях +1		Данные для формы
	1			
	2	Отправить данные в форму	Форма валидна	

Рисунок А.5 – Тест кейсы

Приложение Б

1. Страница входа:

Оцените расположение формы входа на странице	
Насколько информативные предупреждения при ошибочном вводе?	
Оцените оформление страницы входа	
Укажите ваши пожелания и предложения по этому разделу, если они имеются	

2. Страница регистрации:

Оцените расположение формы регистрации на странице	
Насколько информативные подписи к заполнению полей?	
Информативны ли предупреждения при вводе неверных данных?	
Укажите ваши пожелания и предложения по этому разделу, если они имеются	

Рисунок Б.1 – Пример тестового варианта

3. Главная страница:

Оцените цветовое оформление	
Оцените расположение элементов с достопримечательностями	
Оцените расположение фильтров	
Насколько понятен и удобен механизм фильтрации и поиска?	
Укажите ваши пожелания и предложения по этому разделу, если они имеются	

4. Пагинация

Насколько устраивает расположение переключателя страниц	
Насколько устраивает оформление переключателя страниц	

5. Страница достопримечательности

Оцените цветовое оформление	
Оцените текст: шрифты, читаемость	
Оцените фотографии, удобство их переключения	
Оцените удобство работы и визуальную составляющую оценки, возможности ее выставления	
Укажите ваши пожелания и предложения по этому разделу, если они имеются	

Рисунок Б.2 – Пример тестового варианта

6. Страница Мой профиль:

Оцените общее оформление	
Оцените текст: шрифты, читаемость	
Оцените общее расположение элементов	
Укажите ваши пожелания и предложения по этому разделу, если они имеются	

7. Страница создания заявки:

Оцените общее оформление	
Оцените текст: шрифты, читаемость	
<u>Оцените</u> насколько информативны ли предупреждения при вводе неверных данных?	
Укажите ваши пожелания и предложения по этому разделу, если они имеются	

Рисунок Б.3 – Пример тестового варианта