

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

Разработка веб-приложения для бронирования номеров в гостинице

с модулем динамического образования цен на Python

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студентки 3 курса 381 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Кравцовой Анастасии Олеговны

Научный руководитель
доцент, к.ф.-м.н.

И.В. Плаксина

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2023

Введение. За последние несколько лет внутренний туризм в России пережил ряд потрясений. Из-за пандемии изменилось представление россиян об отдыхе и путешествиях. Люди стали чаще выбирать отпуск в родной стране.

Интернет стал наиболее эффективным средством рекламы и продвижения и является одним из важных элементов современной цивилизации. Интернет может удовлетворить все потребности современного человека: это покупки, заключение деловых отношений, поиск клиентов и так далее. Но многие заходят в Интернет читать, развлекаться, общаться, узнавать что-то новое для себя. Поэтому создание сайтов стало таким популярным.

Актуальность темы состоит в том, что гостинице для получения наибольшей выгоды требуются дополнительные способы распространения своих услуг. Сайт - это эффективная площадка для привлечения самой разной целевой аудитории, с помощью которой отель занимается продвижением своих услуг. С его помощью можно предоставлять потенциальным клиентам информацию о ценах, доступных услугах и обеспечить возможность заблаговременно заказать их через модуль онлайн-бронирования.

Необходимо разработать веб-приложение, учитывая потребности потенциальных клиентов туристической отрасли, позволяет улучшить доступность онлайн-бронирования номеров, а динамическое ценообразование способствует увеличению прибыли и эффективному управлению продажами.

Целью работы является проектирование и разработка веб приложения для бронирования номеров в гостинице с модулем динамического ценообразования на языке программирования Python, с использованием фреймворка Django.

Объект исследования — процесс бронирования номеров в отеле через веб-приложение, хранение данных о номерах отеля и возможность добавлять карточки номеров отеля через административную панель, предоставление информации о доступности номеров в гостинице на определенные даты.

Предмет исследования — особенности разработки веб-приложений для бронирования в гостиничном бизнесе, а также формирование цены в зависимости от загрузки.

Для достижения поставленной цели в работе необходимо решить следующие

щие задачи:

- рассмотреть теоретические аспекты, а также методы проектирования и разработки веб-приложений;
- проанализировать предметную область;
- спроектировать веб-приложение и базу данных;
- разработать веб-приложение на языке программирования Python с использованием фреймворка Django.

Практическая значимость работы заключается в улучшении качества обслуживания потенциальных клиентов туристической отрасли, предоставление данных о гостинице, номерах и вариантах размещения для клиента, а также возможность бронирования онлайн, используя динамическое ценообразование.

Магистерская работа состоит из введения, четырех разделов, заключения, списка использованных источников и двух приложений. Во введение описывается актуальность темы работы, указывается цель и задачи работы. В первом разделе описывается роль динамического ценообразования в гостиничном бизнесе, а также рассмотрены основные подходы к ценообразованию. Во втором разделе рассмотрены методы технической реализации задачи. Был сделан выбор модели жизненного цикла, проанализированы структуры веб-приложений и выбрано программное обеспечение для реализации данной задачи. Третий раздел посвящен проектированию веб-приложения и базы данных. В четвертом разделе представлена разработка веб-приложения для бронирования номеров в гостинице с модулем динамического образования цен. В заключении рассматриваются итоги магистерской работы, делаются выводы о проведенной работе. Список использованных источников содержит 21 наименование, на которые в тексте приведены ссылки. В приложениях содержится код диаграмм.

Роль динамического ценообразования в гостиничном бизнесе.

Динамическое ценообразование является моделью управления доходностью в гостиничном бизнесе, которая базируется на непрерывном мониторинге рынка и анализе спроса на предоставляемые услуги. Эта модель позволяет отелям устанавливать цены на номера и дополнительные услуги, в зависимости от разных факторов, таких как ожидаемое количество потребителей, сезон-

ность, конкуренция, спрос и др.

Revenue Management, или управление доходностью — это технология определения лучшей цены для обеспечения высокой доходности гостиницы на основе прогнозирования спроса. Иными словами, управление доходностью — это метод продажи фиксированных запасов нужным клиентам по соответствующей цене в подходящее время. В результате с клиентов могут взиматься разные цены за один и тот же продукт или услугу.

Ключевую роль в динамическом образовании цен играют информационные технологии, позволяющие быстро и точно анализировать данные о спросе, конкурентной среде и других факторах, влияющих на формирование цен. Такие технологии позволяют автоматически корректировать цены на номера и дополнительные услуги, обеспечивая оптимальное соотношение между спросом и предложением.

Итеративная модель. Итеративная модель подразумевает, что процесс разработки разбивается на повторяющиеся циклы, в каждом из которых продукт постепенно совершенствуется. Для итеративной модели не обязательно наличие на старте четко определенного технического задания и требований. Например, заказчик может определить только базовый набор основных функций, а в ходе последующих итераций дополнять их новыми. Отличие от инкрементной модели состоит в том, что в итерационной дорабатывается весь продукт, а не его отдельные блоки. Смысл в том, чтобы результатом каждого цикла была работающая, пусть и не идеальная, модель.

Для разработки веб-приложения была выбрана итеративная модель жизненного цикла, потому что она позволяет быстро адаптироваться к изменяющимся требованиям и обратной связи пользователей, что делает ее отличным выбором для разработки веб-приложений.

Клиент-серверная архитектура веб-приложения. Клиент-серверная архитектура веб-приложения обеспечивает эффективную разработку, развертывание и обслуживание динамичных и интерактивных онлайн-сервисов.

Также существенное преимущество построения веб-приложений для поддержки стандартных функций браузера заключается в том, что функции должны выполняться независимо от операционной системы данного клиента. Вместо того, чтобы писать различные версии для различных операци-

онных систем, приложение создается один раз для произвольно выбранной платформы и на ней разворачивается.

Программное обеспечение и используемые языки программирования. Веб-приложение для бронирования номеров в отеле было принято реализовывать на языке Python и с помощью фреймворка Django.

Python - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Django – свободный программный каркас для веб-приложений на языке Python, использующий шаблон проектирования MVC. Model-view-controller (MVC, «модель-представление-контроллер») – схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области

В качестве СУБД выбрана SQLite. SQLite отлично подойдет для разрабатываемого приложения. Также в Django есть возможность изменять базу данных в процессе роста веб-приложения без потери уже существующих данных. Изменение базы данных происходит с помощью команды миграции.

Создание UML-диаграмм. UML-диаграммы помогают при моделировании архитектуры больших проектов, в которой можно собрать как крупные, так и более мелкие детали и нарисовать каркас (схему) приложения. По нему впоследствии будет строиться код.

Диаграмма вариантов использования показывает взаимодействие между вариантами использования и действующими лицами, отражая функциональные требования к системе с точки зрения пользователя.

Составление вариантов использования производится на основе анализа требований к веб-приложению для бронирования номеров в отеле.

Главными актерами в контексте программного средства являются администратор и пользователь этого приложения.

Пользователи перед бронированием имеют возможность просмотреть разные варианты номеров в отеле. Однако перед тем, чтобы пользователь в полной мере мог совершать какие-либо действия на сайте, ему необходимо зарегистрироваться и авторизоваться.

Прежде чем открывать доступ пользователям, администратору необходимо заполнить каталог номеров отеля. Администратор может создать новую карточку номера, удалить или изменить уже существующий элемент на сайте.

Диаграмма последовательности используется для уточнения диаграмм прецедентов — описывает поведенческие аспекты системы. Диаграмма последовательности отражает взаимодействие объектов в динамике, во времени. При этом информация принимает вид сообщений, а взаимодействие объектов подразумевает обмен этими сообщениями в рамках сценария. В соответствии с рисунком 1 представлена диаграмма последовательности.

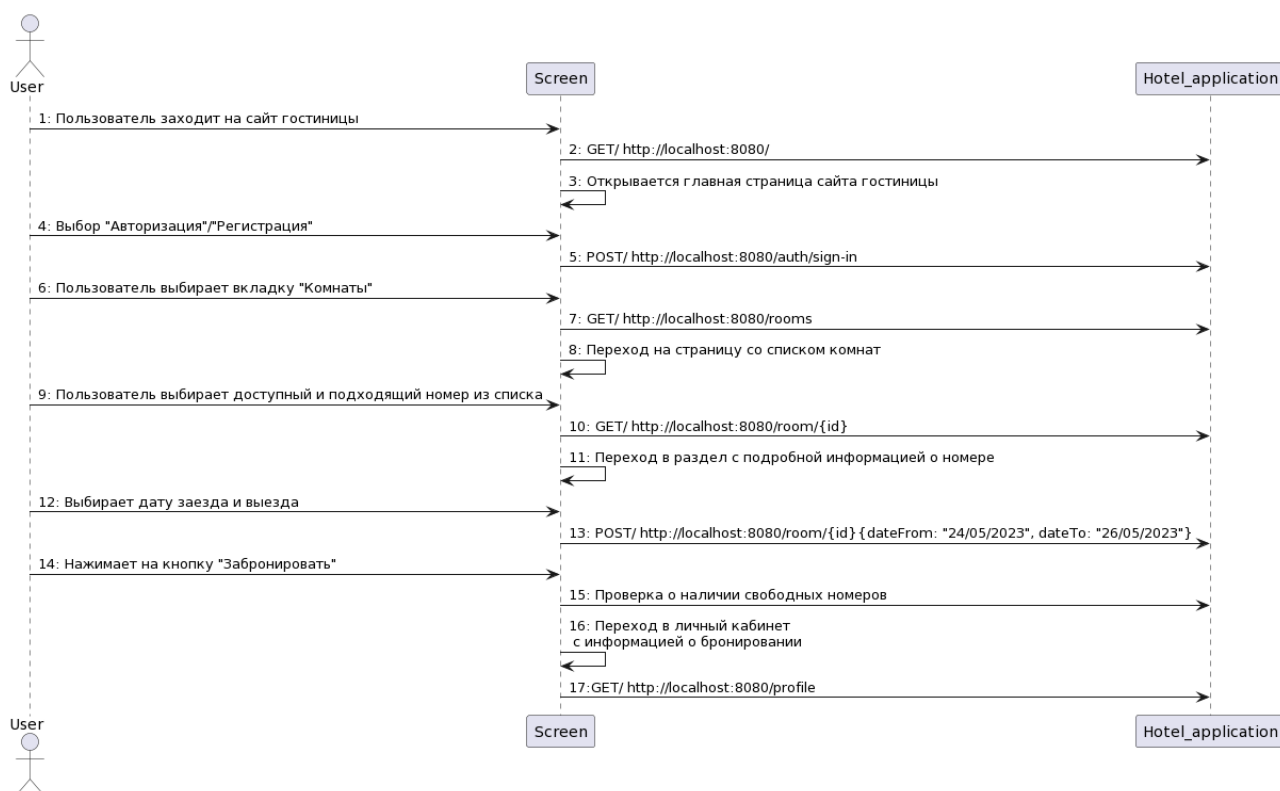


Рисунок 1 - Диаграмма последовательности

Проектирование модели базы данных В сфере разработки программного обеспечения ER-диаграмма, как правило, служит первым шагом в определении требований проекта по созданию информационных систем. На

дальнейших этапах работы ER-диаграммы также применяются для моделирования конкретных баз данных. Реляционная база данных сопровождается соответствующей реляционной таблицей и при необходимости может быть представлена в этом формате.

ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей.

База данных для бронирования номера в отеле через веб-приложение отеля в первую итерацию состоит из 5 сущностей:

1. room
2. app user
3. book data
4. room book data
5. room images

В соответствии с рисунком 2 представлена ER-диаграмма первой итерации.

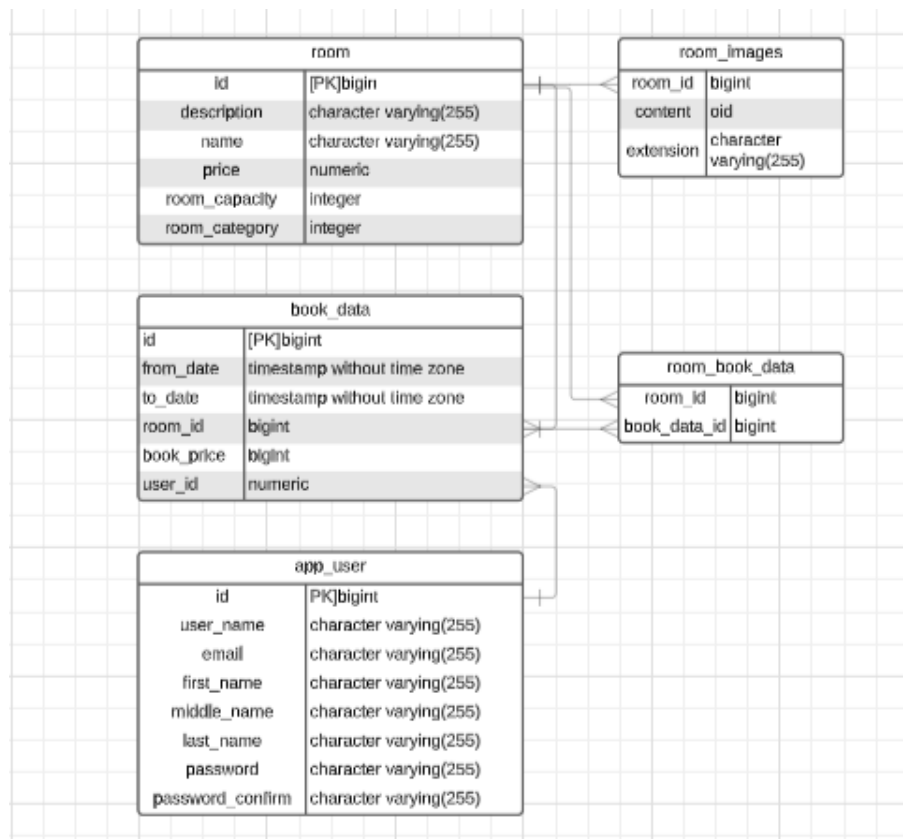


Рисунок 2 - ER-диаграмма первой итерации

Разработка веб-приложения для бронирования номеров в гостинице с модулем динамического образования цен. В качестве среды разработки выбрана оболочка PyCharm, так как она удобна в использовании и имеет бесплатную версию.

Разработка модели данных. В Django таблицы базы данных описываются в виде моделей классов. Модель – это основной источник данных. Он содержит набор полей и поведение данных которые хранятся в базе. Так как Django следует принципу DRY все модели определяются в одном месте.

Django использует принцип ORM, следовательно поддерживаются такие принципы ООП как наследование, полиморфизм и инкапсуляции и т.д. Описание модели данных является аналогом SQL CREATE TABLE.

Созданная в ER-модель базы данных описывается с помощью классов на языке Python. Каждый класс наследует базовый класс Django «Model», который в свою очередь содержит методы по добавлению, удалению и изменению данных. Из ER-модель базы данных следует, что необходимо создать пять классов.

Пример реализации класса Room на языке программирования Python:

1. `class room(models.Model):`
2. `name = models.CharField('Название', max length=50)`
3. `description = models.TextField('Описание', max length=250)`
4. `price = models.TextField('Цена', max length=50)`
5. `room-capacity = models.TextField ('Вместимость', max length=50)`
6. `room-category = models.TextField ('Категория', max length=50)`

Остальные четыре класса описываются также с помощью языка Python.

После создания моделей классов необходимо провести миграцию, чтобы описанные классы появились в базе данных. Миграция осуществляется в терминале с помощью команды: `python manage.py migrate`.

Структура страниц веб-приложения. Интерфейс веб-приложения разрабатывался с использованием HTML, CSS и Python. Содержание каждой страницы сайта определяется HTML кодом, визуализация и оформление — содержанием таблицы стилей CSS, а поведение интерфейса — скриптами Python.

Для упрощения верстки был использован фреймворк Bootstrap — набор

инструментов для создания сайтов и веб-приложений, основанный на современных наработках в CSS и HTML. Он включает в себя HTML и CSS шаблоны оформления для многих компонентов веб-интерфейсов и Python расширения, которые позволяют достаточно просто создавать интерактивные и динамичные интерфейсы.

Базовым шаблоном в веб-приложении Hotel является base.html. Он содержит в себе header и footer сайта, с логотипом. Доступ к этим элементам необходим из любой части сайта, поэтому базовый шаблон всегда будет использоваться при рендеринге страницы. Его наследуют остальные шаблоны.

Шаблоны необходимые для реализации первой итерации:

1. index.html - содержит в себе структуру главной страницы;
2. catalog.html - содержит в себе структуру каталога;
3. room.html - содержит в себе структуру страницы номера в отеле;
4. login.html - содержит в себе структуру страницы авторизации;
5. register.html - содержит в себе структуру страницы регистрации.

Реализация веб-приложения. В ходе работы была создана база данных и прописаны HTML разметки для основных страниц сайта. Далее необходимо прописать функции, которые будут выстраивать логику работы всех страниц сайта. Рассмотрим на примере главной страницы сайта реализацию основных функций приложения, которые обеспечивают взаимодействие пользователя и системы.

Главная страница сайта:

1. def index(request):
2. return render(request, 'Hotel/index.html')

Реализация модуля расчета персональных скидок. Для того чтобы реализовать модуль расчета скидок на бронирование для постоянных клиентов был создан метод класса personal price, который был описан в файле models.py и имеет вид:

```
@classmethod
def personal price(price, booked rooms count):
if booked rooms count < 5:
return price
```

```

if booked rooms count >= 5 and booked rooms count < 7:
return price * 0.97
elif booked rooms count >= 7 and booked rooms count < 10:
return price * 0.95
elif booked rooms count >= 10:
return price * 0.90
return price

```

Метод `personal price` подсчитывает скидку и возвращает новую цену для бронирования в зависимости от количества забронированных ранее комнат клиентом.

Для расчета скидки было задано некоторое правило, в котором скидка зависит от количества бронирований в прошлом:

- Скидка 3% предоставляется пользователю, если он забронировал номер от 5 до 7 раз;
- Скидка 5% предоставляется пользователю, если он забронировал номер от 7 до 10 раз;
- Скидка 10% предоставляется пользователю, если он забронировал номер от 10 и более раз.

Реализация модуля динамического ценообразования. При реализации модуля динамического ценообразования были заданы правила ценообразования, которые зависят от загруженности отеля на определенные дни:

Если загруженность отеля на период бронирования пользователем

1. $\geq 50\%$, то цена за номер увеличивается на 3%,
2. $\geq 70\%$, то цена за номер увеличивается на 5%,
3. $\geq 90\%$, то цена за номер увеличивается на 10%.

Загруженность отеля определяется как отношение проданных номеров в данный период к общему количеству доступных номеров, открытых для продажи в выбранном промежутке времени. Формула расчета загруженности гостиницы:

$$Occupancy(\%) = RoomsSold / RoomsAvailable * 100,$$

где

- Occupancy — это процент занятости номеров в гостинице за определенный период времени,
- Rooms Sold — забронированные номера,
- Rooms Available — доступные номера.

Для реализации модуля динамического ценообразования был создан метод класса `get multiplier`, который был описан в файле `models.py`.

```
@classmethod
def get-multiplier(active-books-count, total-room-count):
    percentage = active-books-count * 100 / total-room-count
    if percentage >= 50 and percentage < 70:
        return Decimal('1.05')
    elif percentage >= 70 and percentage < 90:
        return Decimal('1.1')
    elif percentage >= 90:
        return Decimal('1.2')
    else:
        return Decimal('0.9')
```

В соответствии с рисунком 3 отображается страница номера отеля с выбором дат, в которые бронирование возможно.

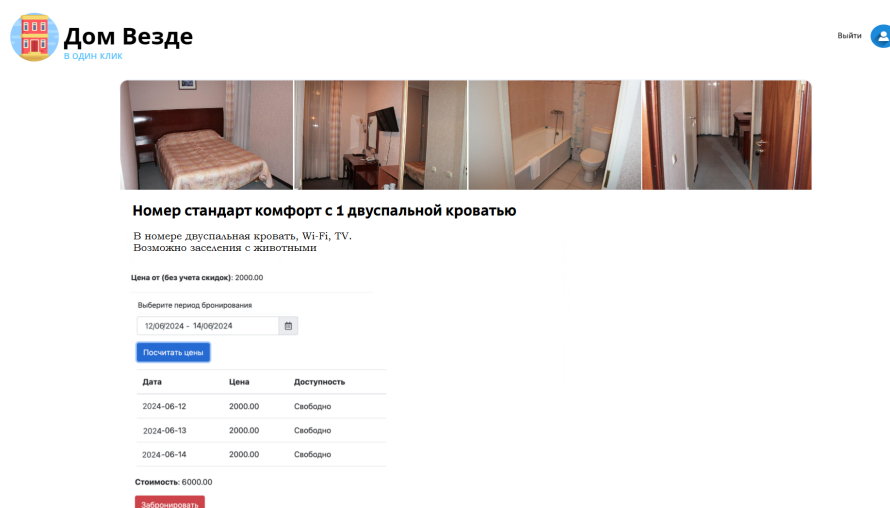


Рисунок 3 - Страница номера отеля с с выбором дат, в которые бронирование возможно

Заключение. В настоящее время большинство туристов пользуются функцией бронирования номеров в гостинице через веб-сайт или приложение, так как это экономит время и позволяет быстро забронировать нужный номер. Веб-приложения для бронирования номеров в гостинице делают это более простым, быстрым и удобным, что увеличивает популярность таких приложений.

Динамическое ценообразование в гостиничном бизнесе — это стратегия установки цен на номера и услуги гостиницы, которая основывается на различных факторах, таких как спрос, сезонность, предложение конкурентов, праздники и другие переменные условия.

Динамическое ценообразование, реализованное в данной работе, является очень важным элементом управления доходностью в гостиничном бизнесе. Вместо традиционного подхода, при котором цены на номера фиксированы и редко меняются, динамическое ценообразование позволяет отелям оптимизировать доходы, учитывая изменения в спросе и других факторах. В результате использования модели динамического образования цен гостиницы могут улучшить свою доходность и обеспечить себе преимущество на рынке. Кроме того, такая модель позволяет отелям более точно прогнозировать доходы и расходы, что упрощает планирование и повышает эффективность управления бизнесом.

В результате написания магистерской работы были достигнуты следующие результаты:

1. проанализирована предметная область;
2. спроектировано веб-приложение и база данных;
3. разработано веб-приложение на языке программирования Python, с использованием фреймворка Django;
4. реализован модуль с динамическим ценообразованием;
5. реализован модуль для расчета персональной скидки клиента.