Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Инженерно-экономический факультет

Кафедра экономической информатики

Дисциплина: Средства и технологии анализа и разработки информационных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему:

WEB-ПРИЛОЖЕНИЕ ДЛЯ ТУРИСТИЧЕСКОЙ СФЕРЫ

Выполнила: гр.773601

Буглак Елена Геннадьевна

Проверил: Сторожев Дмитрий Алексеевич

Минск 2020

# **ВВЕДЕНИЕ**

Современный мир трудно представить себе без использования сайтов. Практически все системы в той или иной степени связаны с функциями долговременного хранения и обработки информации. Фактически информация становится фактором, определяющим эффективность любой сферы деятельности. Современные технологии программирования постоянно развиваются именно в направлении web, давая программистам новые инструменты по разработке именно web-ориентированных приложений.

Благодаря развитию этих технологий появилась возможность автоматизировать многие процессы. Внедрение этих технологий позволит увеличить качество предоставляемых услуг, в которых нуждается заказчик, а по итогу и конечный пользователь.

Туризм представляет собой одну из самых крупных и динамичных отраслей экономики. На различные сектора экономики активно влияют следующие факторы: высокие темпы развития туризма и большие объемы валютных поступлений, которые в свою очередь способствуют формированию собственной туристической индустрии. На зону туризма приходится около 6% мирового валового национального продукта, 7% мировых инвестиций и 11% мировых потребительских расходов. Исходя из этого, можно сделать вывод, что в наши дни индустрия туризма оказывает огромное влияние на мировую экономику.

В данной курсовом проекте объектом исследования рассматривается разработка веб-приложения для туристической сферы, которое позволит эффективно и удобно подобрать тур, также представляет подсистему материальных элементов экономической деятельности и хозяйственных процессов.

Предметом исследования в данном курсовом проекте будет являться построение всех бизнес-процессов, начиная с пользовательских настроек, заканчивая конечными и мелкими настройками для запуска web-приложения.

Ожидается, что автоматизация данной функции приведет к значительному сокращению времени и трудоемкости данного учета, позволит создать единую базу клиентов, а также позволит получать и анализировать отчетность в маркетинговых целях.

При создании приложения для управления всеми необходимыми процессами, нужно учесть множество факторов, для этого в данном приложении должно быть предусмотрено решение следующих задач:

* хранение информацию о различных видах туров;
* реализация функций добавления, удаления тура;
* предоставление пользователю тура в соответствии с его потребностями;
* предоставление пользователю возможности выбора тура, с последующим заказом;
* увеличение клиентской базы.

Целью курсового проекта является улучшение качества предоставления туристических услуг туроператорами и турагентствами путем создания единой рабочей среды для работы с разветвленной сетью партнеров и быстрого доступа к информации о доступности туристических услуг.

Для достижения данной цели необходимо решить следующие задачи:

* описание бизнес-процесса работы компании, предоставляющая тур-поездки (IDEF0);
* функциональное моделирование процессов предметной области;
* разработка алгоритмов работы приложения;
* создание абстрактной системы работы приложения при помощи UML-диаграмм;
* разработка клиент-серверного взаимодействия;
* разработка интерфейса программы для удобной и эффективной работы онлайн-сервиса помощи;
* тестирование программного продукта.
* создание надежной базы данных.

Информационная система позволяет печатать данные о путешествии на типографском бланке - нет необходимости заполнять бланк от руки, намного облегчается ведение и анализ клиентской базы, упрощается работа и расчеты с клиентами. Данные клиента и информация о путешествии вносятся в базу и используются для оформления всех необходимых документов: путевок, заявок, договоров, памяток и т.д. При этом значительно уменьшается вероятность ошибок при заполнении документов (например, ошибок в ФИО туристов или их паспортных данных). Информационная система позволяет легко собирать и анализировать статистические данные о количестве и характере путешествий туристов, качестве работы менеджеров агентства, популярности туристических маршрутов, отелей и т.д.

Что касается ближайшего будущего, то согласно прогнозам этой же Всемирной туристской организации, сфера туризма будет активно развиваться.

Таким образом данный сервис поможет пользователям быстро и удобно выбирать и оформлять покупку тура в зависимости от его потребности и желания.

# **1 ОПИСАНИЕ ТУРИЗМА**

Туризм — это временные выезды (путешествия) людей в другую страну или местность, отличную от места постоянного жительства на срок от 24 часов до 6 месяцев в течение одного календарного года или с совершением не менее одной ночевки в развлекательных, оздоровительных, спортивных, гостевых, познавательных, религиозных и иных целях без занятия деятельностью.

Туризм как массовое социальное явление стал складываться лишь после Второй мировой войны, хотя корни туризма уходят в глубокое прошлое. В истории развития туризма различают четыре этапа:

Первый этап – от древности до начала XIX века.  
Второй этап – от начала XIX века до начала XX века.  
Третий этап – от начала XX века до Второй мировой войны.  
Четвертый этап – после Второй мировой войны до наших дней.

Четвертый этап называется этапом массового туризма. Именно в этот период туризм приобретает массовый характер. Из предмета роскоши он становится потребностью для большинства населения индустриально развитых стран. Формируется индустрия отдыха и развлечений со своими институтами, продуктом, производственным циклом, методами организации и управления производством.

Классификация туризма — выделение внутренне однородных таксонов туристской деятельности по принятым основаниям. Наиболее распространенной классификацией туризма является его деление на типы, категории, виды и формы.

Тип туризма определяется национальной принадлежностью туристов. Всемирная туристская организация (ВТО) предложила классификацию туризма по следующим типам: внутренний, въездной (активный) и выездной (пассивный). Основным критерием классификации является пересечение государственных границ. Внутренний туризм — путешествия по стране лиц, постоянно проживающих в ней; выездной туризм — путешествия лиц, постоянно проживающих в государстве, в другую страну. Въездной туризм — путешествия по своей стране лиц, проживающих постоянно в других странах. Указанные типы туризма по-разному сочетаются между собой, образуя категории туризма.

Категории туризма включают международный и национальный туризм, а также туризм в пределах страны. С точки зрения приведенной классификации туризма по типам к национальному туризму относится внутренний и выездной туризм, к международному туризму — въездной и выездной. Туризмом в пределах страны занимаются как граждане данной страны, так и приезжающие из других стран и регионов. При этом ночующие посетители считаются туристами, а однодневные посетители — экскурсантами. В качестве экскурсантов выступают и туристы, прибывшие в страну на круизном судне, если они не используют для ночевки местные средства размещения.

Различают следующие основные сферы турбизнеса:

- производство;

- коммерция;

- финансовое предпринимательство;

- консалтинг.

Такое деление является условным, поскольку часто очень сложно разграничить производственный и непроизводственный бизнес. Если кратко определить важность каждой сферы в туризме и национальной экономике, то можно сказать следующее: в разное время каждая из этих сфер может давать различный доход, однако главной среди них остается производство. Именно в сфере производства создаются материальные блага (туристические продукты и услуги), необходимые для жизнедеятельности человека. Да, коммерция в турбизнесе существует только потому, что туроператор создал продукт, который турагент перепродает. Состояние туризма как особого межотраслевого комплекса экономики, определяется количеством высококачественных туристических продуктов и услуг, которые приходятся на одного человека. Эти же продукты и услуги в составе ВВП страны влияют на уровень развития национальной экономики. Весомой по количеству зарегистрированных туристических организаций, сейчас есть доля коммерции. Это объясняется прежде всего относительно низким стартовым капиталом и конкурентоспособной рентабельностью бизнеса, однако по уровню доходов доминирует производство.

В 2010 году Республику Беларусь с деловыми, туристическими и служебным целями посетили 4 млн. иностранных граждан, экспорт туристических услуг составил 146,7 млн. долларов США, что в 6,8 раза больше, чем за аналогичный период 2005 года. Объем платных туристических и экскурсионных услуг, оказанных населению, составил 468,7 млрд. рублей и увеличился в фактических ценах в 13 раз по сравнению с 2005 годом. В настоящее время эксперты не могут точно сказать, как будет развиваться турбизнес в Республике Беларусь в связи с короновирусной пандемией. На основании изложенного целью развития туризма в Беларуси является формирование современного, эффективного и конкурентоспособного туристского рынка, обеспечивающего широкие возможности для удовлетворения потребностей как белорусских, так и иностранных граждан в туристских услугах, повышение занятости и уровня доходов населения.

Сравнив наш сервис с booking.com и с airbnb.ru, можно заметить, что необходимый функционал нашего приложения будет меньше, однако основная функция и цель его будет такой же.

# **2 ОПИСАНИЕ ОСНОВНЫХ ПРОЦЕССОВ РАБОТЫ СЕРВИСА ПО УПРАВЛЕНИЮ БРОНИРОВАНИЕМ ТУРОВ**

Моделирование бизнес-процессов нужно проводить для выявления слабых и уязвимых мест компании, так же для повышения эффективности её работы.

Анализ начинается с построения модели как есть (AS-IS), которая отражает положение дел организации на текущий момент, позволяет систематизировать протекающие в ней процессы и информационные потоки в рамках этих процессов.

Данный метод применяется для определения процессов, которые подлежат автоматизации, хотя может применяться и только для организационных изменений.

Моделирование процессов AS-IS позволяет: определить набор действий, составляющих процесс и определить порядок выполнения действий в рамках бизнес-процесса; произвести разделение зон ответственности; определить ресурсы, потребляемые в ходе выполнения бизнес-процесса; определить степень взаимодействия между участвующими в процессе сотрудниками и подразделениями; определить информационные потоки в рамках исследуемого процесса, включая движение документов в бумажной или электронной форме; определить потенциально узкие места и возможности для улучшения процесса, которые будут использованы для его оптимизации в дальнейшем.

После исправления недостатков бизнес-модели AS-IS происходит переход к модели TO-BE (как должно быть). На основе данной модели начинается процесс проектирования модели данных, а также информационной системы.

Построим модель AS-IS на основе «очного» туристического агентства и модель TO-BE на основе внедрения онлайн-системы.

## **2.1 Описание модели «как-должно-быть»**

На рисунке 2.7 представлена Контекстная диаграмма модели TO-BE.

Декомпозиция верхнего уровня контекстной диаграммы представлена на рисунке 2.8. Она разбивается на четыре блока: «Войти как пользователь», «Войти как администратор», «Войти как туроператор», «Просмотреть сайт анонимно».

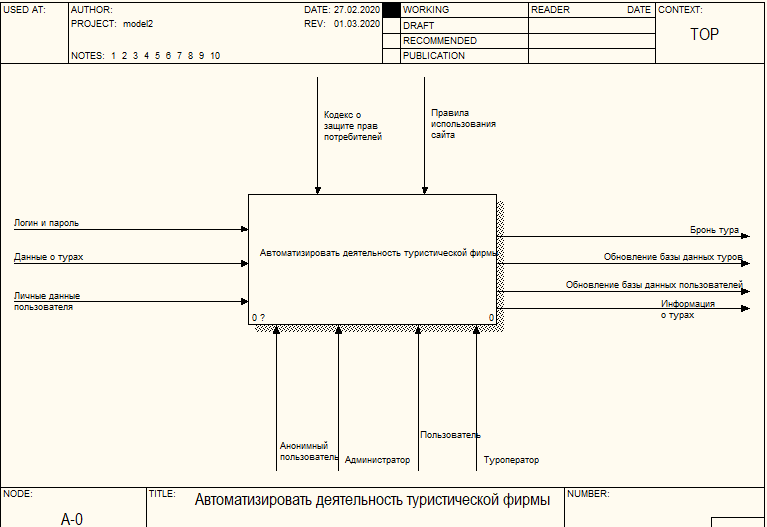


Рисунок 2.7 – Контекстная диаграмма автоматизации деятельности туристической фирмы

На рисунке 2.8 представлена декомпозиция основного блока. Ее результатом являются следующие уровни: «Войти как пользователь», «Войти как администратор», «Войти как туроператор» и «Просмотреть сайт анонимно». Прежде чем начать разработку сайта, первое, что необходимо сделать это продумать структуру: ключевые разделы, возможности для посетителей. Определяется функционал сайта. Все этапы подразумевают составление списка необходимых элементов сайта, определение перечня технологий, продумывается список как внешних функций сайта (для посетителей), так и внутренних. Далее веб-дизайнер разрабатывает дизайн. После выше перечисленных этапов программист приступает к разработке самой программы, а по окончании контент-менеджер добавляет контент.

Результатом декомпозиции блока «Войти как пользователь» (рисунок 2.9) являются следующие уровни: «Выбрать тур», «Забронировать тур», «Оставить комментарий», «Изменить личные данные».

Результатом декомпозиции блока «Войти как администратор» (рисунок 2.10) являются следующие уровни: «Просмотреть туры», «Редактировать учетную запись пользователя», «Удалить пользователя», «Назначить роль пользователя», «Изменить настройки аккаунта».

Результатом декомпозиции блока «Войти как туроператор» (рисунок 2.11) являются следующие уровни: «Работать с турами», «Одобрить бронь», «Отклонить бронь», «Ознакомиться с турами», «Редактировать личный аккаунт».

Результатом декомпозиции блока «Просмотреть сайт анонимно» (рисунок 2.12) являются следующие уровни: «Просмотреть туры», «Ознакомиться с комментариями», «Зарегистрироваться».

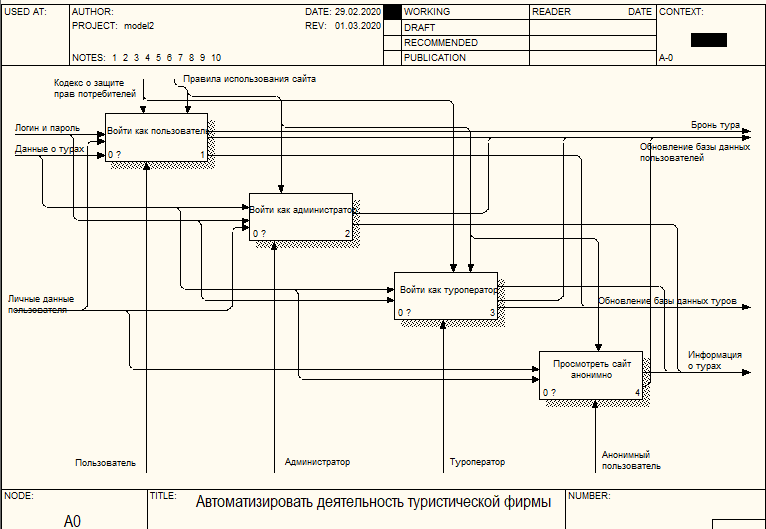


Рисунок 2.8 – Декомпозиция верхнего уровня контекстной диаграммы

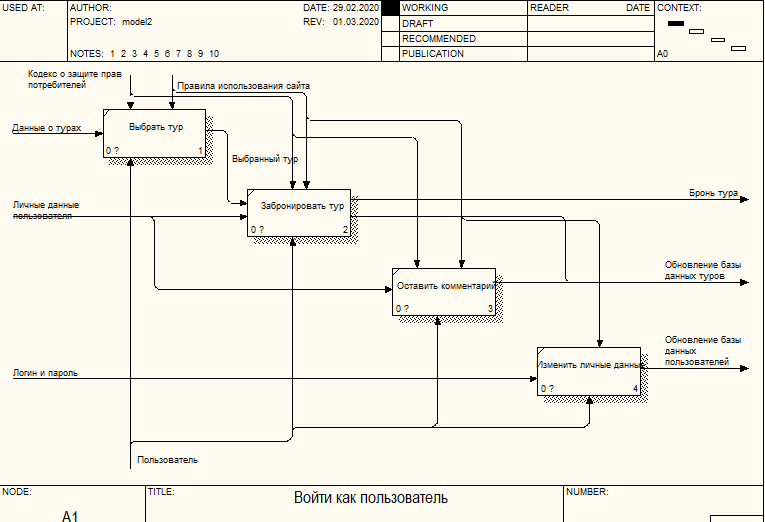


Рисунок 2.9 – Декомпозиция блока «Войти как пользователь»

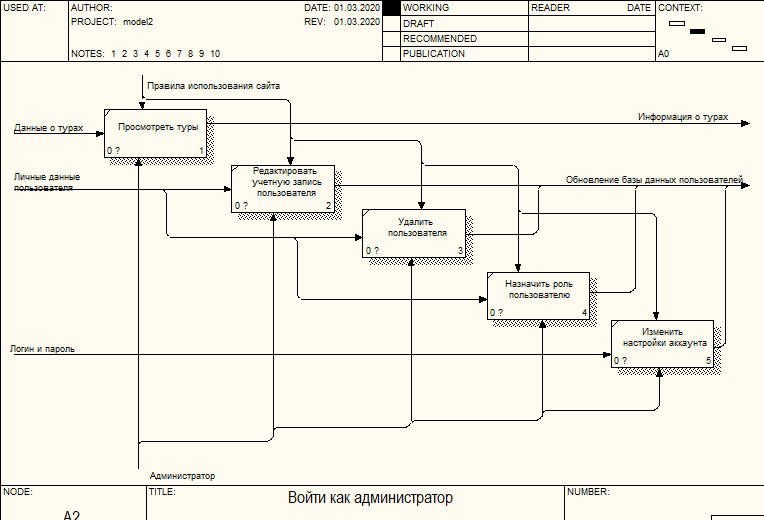


Рисунок 2.10 – Декомпозиция блока «Войти как администратор»

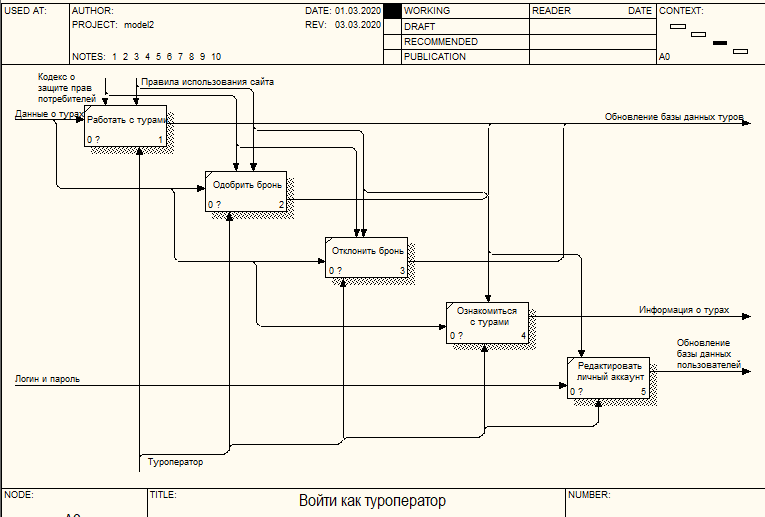


Рисунок 2.11 – Декомпозиция блока «Войти как туроператор»

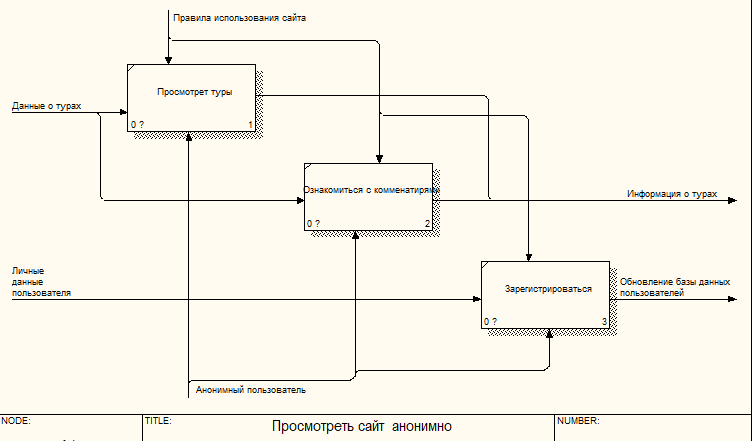


Рисунок 2.11 – Декомпозиция блока «Просмотреть сайт анонимно»

# **3 СПЕЦИФИКАЦИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ**

Суть диаграммы вариантов использования состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

Для разрабатываемой системы была спроектирована диаграмма вариантов использования (рисунок 3.1).

В данном проекте используются три действующих лица: пользователь, администратор и компания, но, для понятности и удобности, диаграмма разделена нас две части и здесь представлены действия пользователя и туроператора. Туроператор представляет свою компанию на данной площадке. Ее основными задачами являются управление турами и общение с потенциальными клиентами.

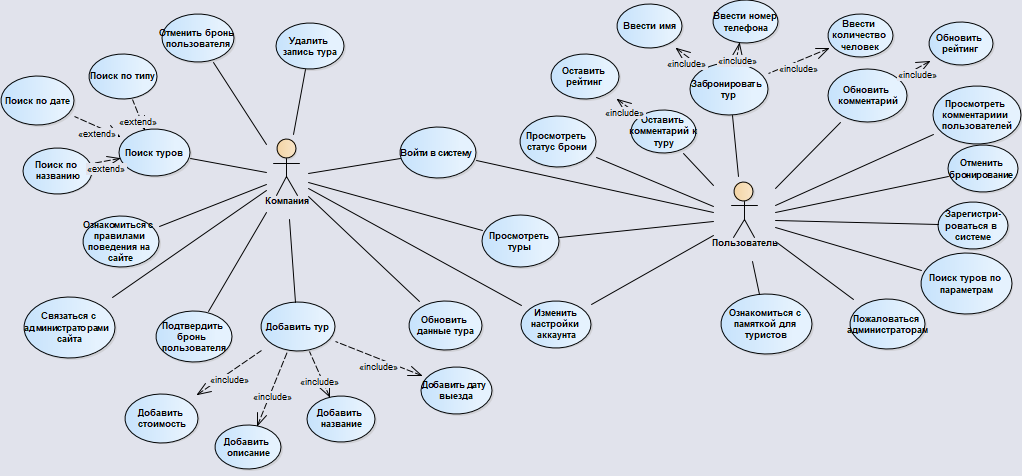


Рисунок 3.1 – Диаграмма вариантов использования

# **4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЁ ОПИСАНИЕ**

Для реализации web-приложения для туристической схемы была разработана информационная модель, на основании которой строилась база данных (Рисунок 4.1)

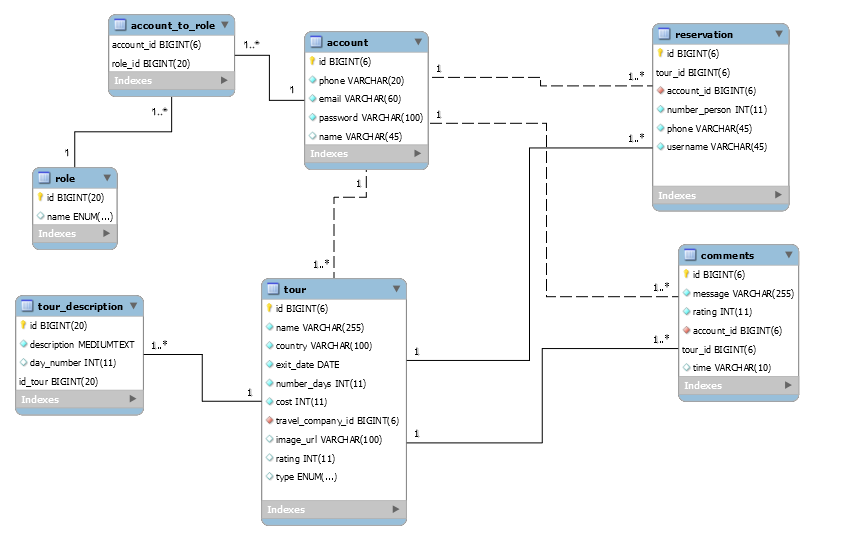


Рисунок 4.1 – Информационная модель web-приложения для туристической сферы

Процесс проектирования базы данных с использованием метода нормальных форм является итерационным и заключается в последовательном переводе отношения из первой нормальной формы в нормальную форму более высокого порядка по определенным правилам. Каждая следующая форма ограничивается определенным типом функциональных зависимостей и устранением соответствующих аномалий при выполнении операций над отношениями базы данных, а также сохранении свойств предшествующих нормальных форм.

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц). Цель нормализации: исключить избыточное дублирование данных, которое является причиной аномалий, возникших при добавлении, редактировании и удалении строк таблицы.

Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

Рассматриваемая база данных соответствует первой нормальной форме, так как в ней присутствуют только уникальные записи.

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от Первичного Ключа(ПК).

Рассматриваемая база данных соответствует второй нормальной форме, так как она спроектирована так, что каждая таблица содержит только те поля, значения которых могут зависеть только от параметров собственной таблицы и никак не зависят от полей других таблиц.

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут не транзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы, что и сделано в проектируемой базе данных.

# **5 ОБОСНОВНИЕ ВЫБОРА КОМПОНЕНТ И ТЕХНОЛОГИЙ ДЛЯ РЕАЛИЗАЦИИ КУРСОВОГО ПРОЕКТА**

## **5.1 Описание выбранных компонентов технологий и обоснование их выбора**

При написании web-приложения использовались следующие компоненты и технологии: язык Java, Apache Maven, Tomcat, Spring Core, Spring MVC, Spring Security, Hibernate, HTML, CSS, Bootstrap, jQuery, ajax.

Для написания веб-приложения был выбран язык Java. Причиной этому являются некоторое особенности языка. Во-первых, в отличие от многих других языков, включая C и C++, Java, когда был создан, он не компилировался в платформе конкретной машины, а в независимом от платформы байт-коде. Этот байт код распространяется через интернет и интерпретируется в Java Virtual Machine (JVM), на которой он в настоящее время работает. Во-вторых, Java прилагает усилия, чтобы устранить ошибки в различных ситуациях, делая упор в основном на время компиляции, проверку ошибок и проверку во время выполнения. В-третьих, в Java встроено огромное количество мощных стандартных библиотек. В-четвертых, у Java есть встроенная поддержка работы в сетях (как локальных, так и Internet). В-пятых, у Java есть функции многопоточности, можно писать программы, которые могут выполнять множество задач одновременно. Введение в язык Java этой конструктивной особенности позволяет разработчикам создавать отлаженные интерактивные приложения. В-шестых, введение Just-In-Time компилятора позволило получить высокую производительность.

Для автоматизации сборки проектов использовался фреймворк Apache Maven. При этом сборка проектов осуществляется на основе описания структуры и зависимостей в файле pom.xml.

Tomcat позволяет запускать [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и содержит ряд программ для самоконфигурирования, используется в качестве самостоятельного [веб-сервера](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), в качестве сервера контента.

Для разработки web-приложения использовался фреймворк Spring MVC. Фреймворк Spring MVC обеспечивает архитектуру паттерна Model-View-Controller при помощи слабо связанных готовых компонентов. Паттерн MVC разделяет аспекты приложения (логику ввода, бизнес-логику и логику UI), обеспечивая при этом свободную связь между ними. В приложении это реализовано тем, что его работа разделена на три составляющие: пакет controller, пакет service и пакет views.

Http-запрос, который приходит со страницы, обрабатывается классами-контроллерами, которые в свою очередь передают управление дальше сервисами, который будут обращаться через репозитории к базе данных. Ответом от контроллера является либо jsp-страница, либо ResponseEntity, которая содержит какой-либо объект, а также http-статус, который является ответом контроллера и результатом обработки запроса.

Spring Security это Java/JavaEE framework, предоставляющий механизмы построения систем аутентификации и авторизации, а также другие возможности обеспечения безопасности для корпоративных приложений, созданных с помощью Spring Framework.

В разрабатываемом приложении с помощью Spring Security реализована авторизация пользователей, а также контроль доступа к ресурсам, с помощью класса конфигураций.

Hibernate – самая популярная реализация спецификации JPA, предназначенная для решения задач объектно-реляционного отображения (ORM). Целью Hibernate является освобождение разработчика от значительного объёма сравнительно низкоуровневого программирования при работе в объектно-ориентированных средствах в реляционной базе данных. Исходя из этого, способом работы с базой данных был выбран Hibernate.

Для создания веб-страниц использовались такие языки, как HTML, CSS, Bootstrap, JS и его фреймворки.

HTML – это гипертекстовый язык разметки, который используется для создания документов в Интернет(веб-страниц). При помощи HTML создавалась веб-страница, содержащая все нужные элементы.

CSS (каскадные таблицы стилей) — это язык, который отвечает за визуальное представление документов пользователю. Стили для элементов веб-страницы можно прописывать как в самом html-документе, так и в отдельном файле. В проекте используется внешний файл, который при помощи специального тега подключается к html-документу. Такие файлы имеют расширение \*.css.

Для упрощения создания разметки страницы использовались готовые шаблоны Bootstrap.

Bootstrap — свободный набор инструментов для создания [сайтов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82) и [веб-приложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5). Включает в себя [HTML](https://ru.wikipedia.org/wiki/HTML)- и [CSS](https://ru.wikipedia.org/wiki/CSS)-шаблоны оформления для [типографики](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0), веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая [JavaScript](https://ru.wikipedia.org/wiki/JavaScript)-расширения.

jQuery — [набор функций JavaScript](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_JavaScript), фокусирующийся на взаимодействии [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) и [HTML](https://ru.wikipedia.org/wiki/HTML). Библиотека jQuery помогает легко получать доступ к любому элементу [DOM](https://ru.wikipedia.org/wiki/Document_Object_Model), обращаться к атрибутам и содержимому элементов [DOM](https://ru.wikipedia.org/wiki/Document_Object_Model), манипулировать ими. Также библиотека jQuery предоставляет удобный [API](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) для работы с [AJAX](https://ru.wikipedia.org/wiki/AJAX).

AJAX (аббревиатура от «Asynchronous Javascript And Xml») – технология обращения к серверу без перезагрузки страницы. За счёт этого уменьшается время отклика и веб-приложение по интерактивности больше напоминает десктоп.

JQuery и AJAX в проекте используются для взаимодействия веб-страниц с контроллерами. С их помощью формируются запросы на страницах.

## **5.2 Диаграмма последовательности для функции «Авторизация»**

Самой богатой и гибкой формой диаграмм взаимодействий является диаграмма последовательностей. Она представляет взаимодействие между линиями жизни как упорядоченную последовательность событий.

Рассмотрим диаграмму последовательностей, представленную на рисунке 5.1.

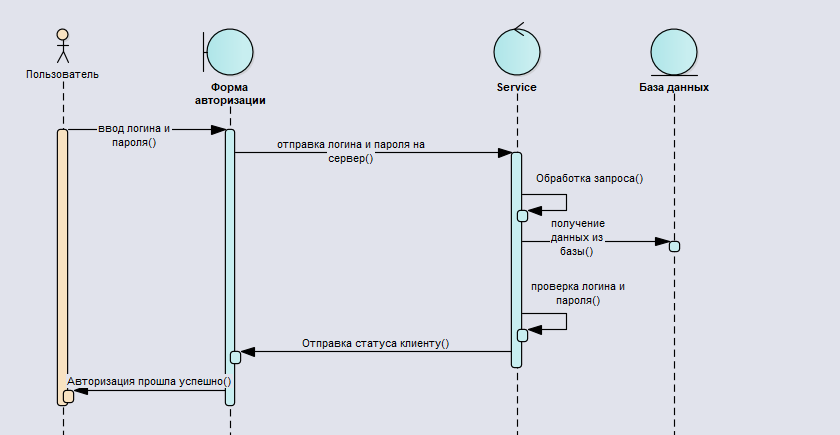


Рисунок 5.1 – Диаграмма последовательности для процесса авторизации пользователя

Диаграмма последовательностей отображает временные особенности передачи и приема сообщений между объектами. Время на диаграмме развивается сверху вниз, а жизненный цикл горизонтально. Фокус управления первоначально находится на актере «Пользователь». На странице авторизации он входит в систему, вводя при этом логин и пароль. Введенные данные отправляются на сервер. Обрабатывается запрос. В случае успешного ввода, сервер проверяет логин и пароль на совпадение в базе данных и возвращает ввод пользователя обратно сервису. Сервис возвращает в форму авторизации статус, а пользователь видит исход авторизации.

## **5.3 Диаграмма компонентов**

Диаграмма компонентов (component diagram) ‒ показывает взаимосвязи между модулями (логическими, физическими), из которых состоит моделируемая система. Основной тип сущностей на диаграмме компонентов – компоненты и интерфейсы, посредством которых указывается взаимосвязь между компонентами. На диаграмме компонентов применяются следующие отношения:

1. реализации между компонентами и интерфейсами (компонент реализует интерфейс);
2. зависимости между компонентами и интерфейсами (компонент использует интерфейс).На рисунке 5.2 представлена диаграмма компонентов, которая обуславливает архитектуру разрабатываемой системы.

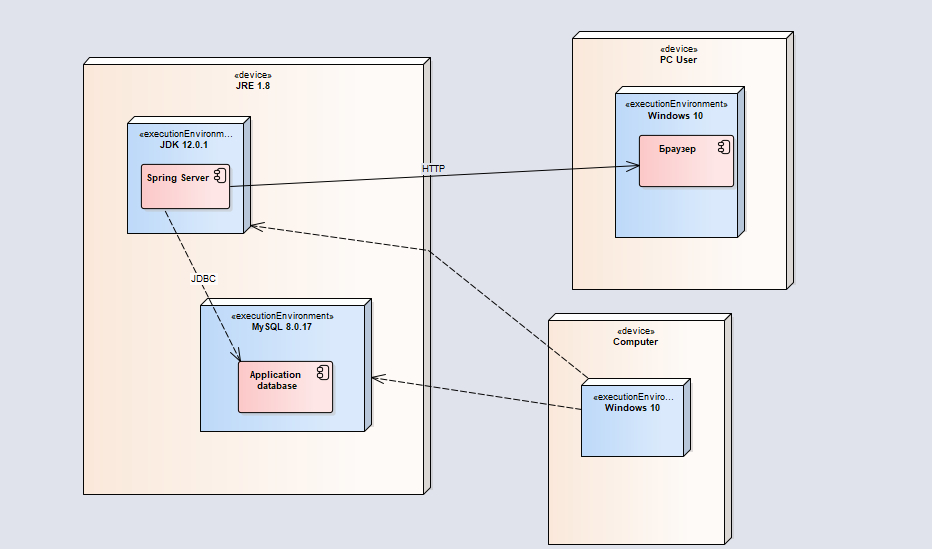


Рисунок 5.2 – Диаграмма компонентов

## **5.4 Диаграмма развертывания**

Диаграмма размещения (deployment diagram) наряду с отображением состава и связей элементов системы показывает, как они физически размещены на вычислительных ресурсах во время выполнения. Таким образом, на диаграмме размещения, по сравнению с диаграммой компонентов, добавляется два типа сущностей: артефакт, который является реализацией компонента и узел (может быть как классификатор описывающий тип узла, так и конкретный экземпляр), а также отношение ассоциации между узлами, показывающее, что узлы физически связаны во время выполнения.

Процесс создания диаграммы состоит из двух этапов. Первый, основное внимание уделяется на узлы и соединения. Второй, распределение экземпляров артефактов по экземплярам узлов или же артефактов по узлам. [4].

Рассмотрим созданную диаграмму развертывания системы (Рисунок 5.3).

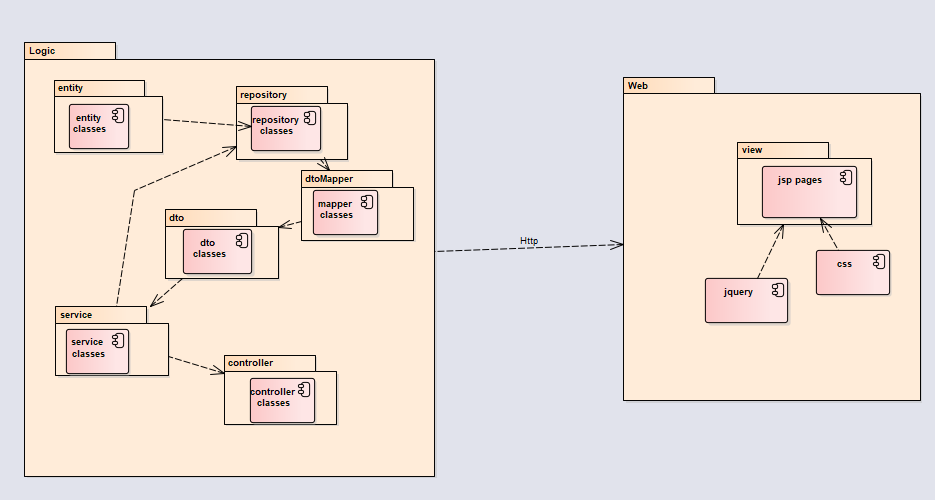


Рисунок 5.3 – Диаграмма развертывания

**6 МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ И ИХ ОПИСАНИЕ**

Для визуального моделирования систем используется универсальный язык моделирования UML. Основной идеей данного языка является возможность моделирования ПО и других систем как наборы взаимодействующих объектов. В UML существует три строительных блока, такие как сущности, отношения, диаграммы. Сущности представляют элементы модели. Отношения определяют связь между сущностями. Диаграммы показывают наборы сущностей, которые являются способом представления того, что будет делать система или как. [4].

**6.1 Диаграмма состояния для варианта использования «Регистрация»**

**Диаграмма состояний (state machine diagrams)** – это известная технология описания поведения системы, которая показывает поведение одного объекта в течение его жизни.

Ниже представлена диаграмма, которая описывает процесс регистрации пользователя в системе. Для этого ему нужно ввести исходные данные. Если они будут удовлетворять условиям, то новая учетная запись будет создана и пользователю будет предложено авторизоваться в системе.

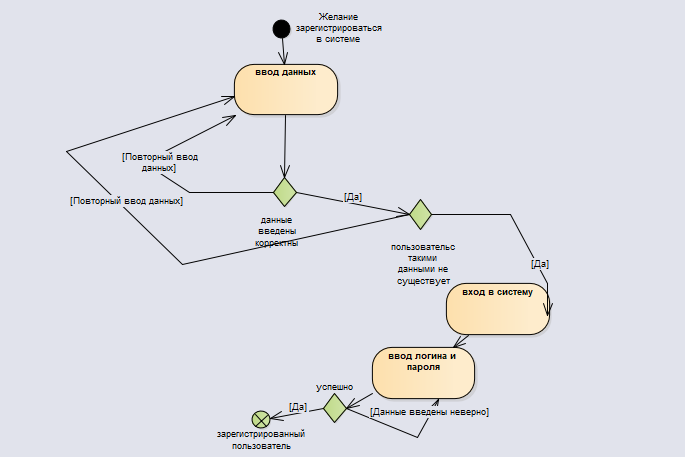


Рисунок 6.1 – Диаграмма состояний

**6.2 Алгоритм регистрации пользователя**

# **8 РУКОВОДСТВО ПО РАЗВЕРТЫВНИЮ СИСТЕМЫ**

Для запуска приложения необходимо установить MySQL, JDK версии 1.8 и выше, а также JBoss EAP.