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

Выполнили: гр.773601

Манкевич Данила Андреевич

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.

В данной курсовом проекте рассматривается разработка веб-приложения для туристической сферы, которое позволит эффективно и удобно заказывать билеты в тур.

При создании приложения для управления процессами купли билетов 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 данного курсового проекта является разработать web-приложения для туристической сферы понятное пользователю. Данный онлайн-сервис позволит быстро подобрать и купить необходимый тур либо просто посмотреть предложения на сайте с целью ознакомления.

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.

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

# **1 ОПИСАНИЕ СЕРВИСА ПО ПОКУПКЕ ТУРА**

Под туром в любую точку мира следует понимать то, что в силу своих естественных особенностей они могут быть удалены или изменены воздействием человека. Обычного покупателя не должно это волновать, он обязан получить то, что хочет и за что будет платить. Для этого необходимо обеспечить и держать с ним связь, информировать и предоставлять необходимую информацию, чтоб у него не было никаких претензий.

Каждому человеку необходим отдых, следовательно, перед каждым рано или поздно стаёт вопрос о приобретении тура для отдыха. Хотя это может быть и не только через туристическую организацию, но и самовольный переезд без посредников. Хоть предоставление услуг по продажам туров нужны не каждому, всё же спрос на неё довольно велик.

В последнее время увеличивается количество людей, готовых получить готовую тур-поездку, в частности, с отелем, переездом, экскурсиями и многим другим, которые готовы переплатить, чтоб обеспечить себе хороший отдых. Обострился интерес и к самому просмотру тех мест, куда бы человек захотел бы поехать без участия тур-операторов в будущем. В этой сфере возникает много острых, дискуссионных проблем, которые представляют большой интерес. Рынок туристических организаций является не только динамичным и бурно развивающимся, но также и не очень стабильным. С развитием огромного количества ресурсов для аренды и транспортировки в отдельности и за более мелкий прайс, обычные люди рассматривают данный вариант. Чтоб заинтересовать пользователя, нужно стабильно, качественно и без перебоев работать.

Говоря о сделках с туристическими агенствами нужно провести черту между желаемым и полученным, если оба понятия совпадут и будут идентичны, тогда цель агенства будет вознаграждена отзывом, ценой и возможной рекомендацией для других людей. Отношения, связанные между человеком и оператором регулируются договором и законом о защите прав потребителя.

Следует отметить, что современный рынок предложений по турам отличается от рынка прошлых лет. Теперь выбор мест увеличен и качество услуг из-за высокой конкуренции увеличен. В сделках по аренде мест в отелях и взаимоотношении с компаниями, занимающимися транспортировкой людей большую роль играет престиж компании, отзывы и многое другое, поэтому необходимо соблюдать все требуемые условия, связанные с договором.

Учитывая высокий спрос и не всегда высокую стоимость туров, сумма заключаемых сделок оценивается ситуационно, но главное с получением в конечном результате прибыли. Тур-бизнес сейчас является одним из самых выгодных, так как по сути в данной сфере не нужно что-либо производить, а брать только самое лучшее и перепродавать за более высокую стоимость.

Из всего выше сказанного можно сделать следующие выводы: если данный сервис позволит покупателям максимально быстро и с минимальными рисками приобретать тур, то агенство будет пользоваться популярностью. Но почему именно онлайн-сервис? Потому что это позволяет выполнять покупки быстро и без лишних хлопот и затрат времени, пользователь может выбирать то, что он хочет или ему понравится. Открыть онлайн-сервис намного быстрее и дешевле чем арендовать офис нанимать людей и открывать предприятие. Онлайн сервис доступен постоянно – 24 час в сутки и 7 дней в неделю в отличии от обычной компании. Также он охватывает куда большую аудиторию, ведь клиенту не придётся ехать в офис компании, он всё что ему нужно выбрать и оплатить из дома. Ещё одним плюсом онлайн-сервиса является возможность автоматизации практически всех процессов, следовательно, существенное сокращение расходов на обслуживание и персонал. Так же одним из важнейших плюсов является возможность аналитики. Можно отслеживать практическое каждое действие пользователя на странице, анализировать эффективность каждого добавленного модуля и его целесообразность.

# **2 ОПИСАНИЕ ОСНОВНЫХ ПРОЦЕССОВ РАБОТЫ СЕРВИСА ПО УПРАВЛЕНИЮ САЙТА**

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 агенства и модель TO-BE на основе внедрения онлайн-системы .

## **2.1 Описание модели понятным языком для заказчика**

В данной работе была разработана диаграмма IDEF0. Стандарт IDEF0 предназначен для функционального моделирования.

На рисунке 2.1 представлена Контекстная диаграмма оптимизации процесса работы оформления покупки в тур-салоне, а также определены потоки входных и выходных данных, механизмы ограничения и управления данными. Входными данными для основного процесса являются запросы данные об актуальном ассортименте и подготовленные клиентом документы. Функционирование магазина без клиентов и менеджера невозможно. Механизмом управления выбраны клиент и менеджер. Чтобы организовать все на законном основании были выбраны следующие правила: Законодательство РБ и защита прав потребителя. Результатом деятельности являются: Заключенный договор и отчет о продаже.

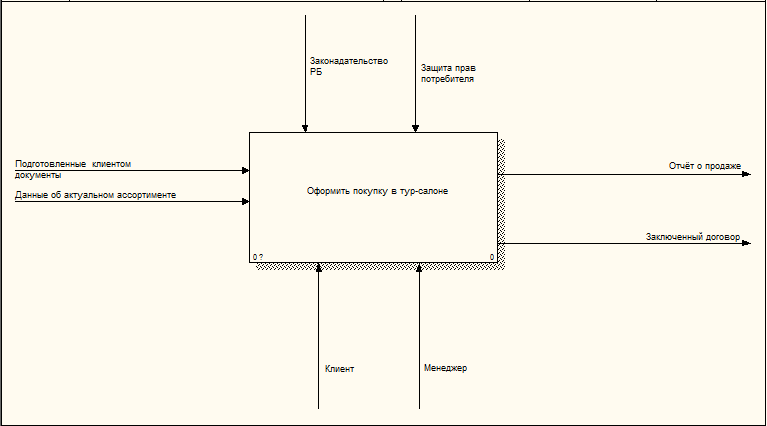


Рисунок 2.1 – Контекстная диаграмма процесса работы туристического агентства

Декомпозиция верхнего уровня контекстной диаграммы представлена на рисунке 2.2. Она разбивается на три блока: «Выбрать тур», «Оформить заказ», «Заключить договор купли-продажи».

Проанализируем каждый уровень.

Результатом декомпозиции блока «Выбрать тур» (рисунок 2.3) являются следующие уровни: «Выбрать категорию тура», «Выбрать место поездки», «Выбрать переезд», «Выбрать время поездки».

Результатом декомпозиции блока «Оформить заказ» (рисунок 2.4) являются следующие уровни: «Заказать отель и переезд», «Оформить дополнительные услуги», «Оплатить тур».

Здесь можно предоставить как будет происходить оплата тура и результатом декомпозиции блока «Оплатить тур» (рисунке 2.5), где можно увидеть следующие уровни: «Оформить рассрочку», «Оформить кредит», «Оплатить одним платежом».

Результатом декомпозиции блока «Заключить договор купли-продажи» (Рисунок 2.6) являются следующие уровни: «Заполнить личные данные», «Заполнить паспортные данные», «Проверить оплату», «Оформить гарантию», «Заключить договор».

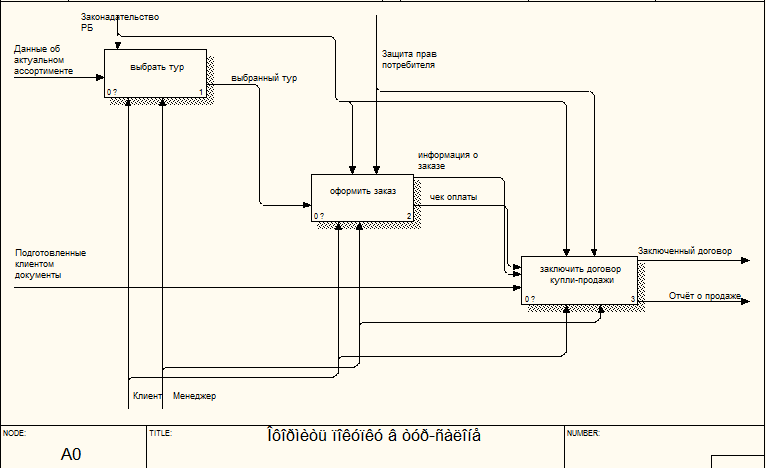


Рисунок 2.2 – Декомпозиция верхнего уровня контекстной диаграммы

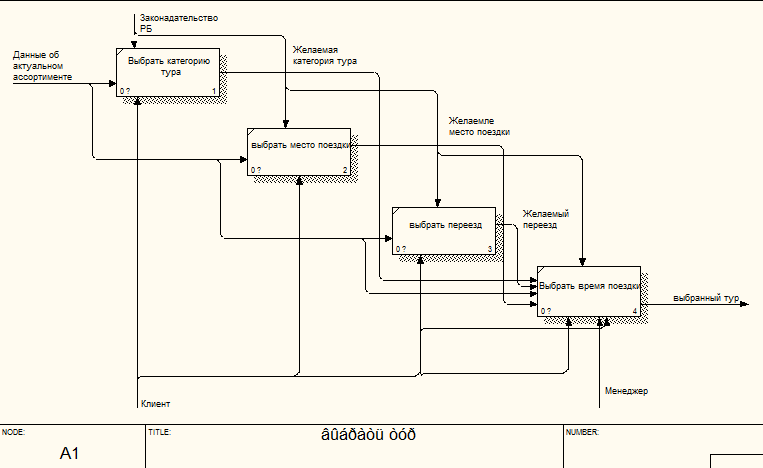


Рисунок 2.3 – Декомпозиция компонента «Выбрать тур»

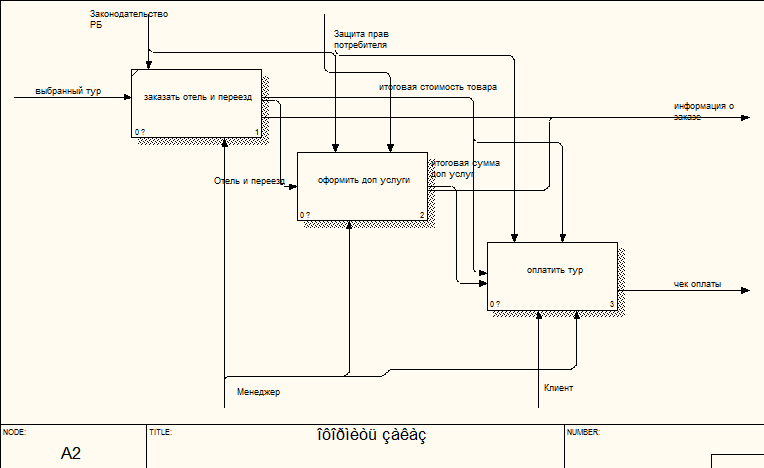


Рисунок 2.4 – Декомпозиция компонента «Оформить заказ»

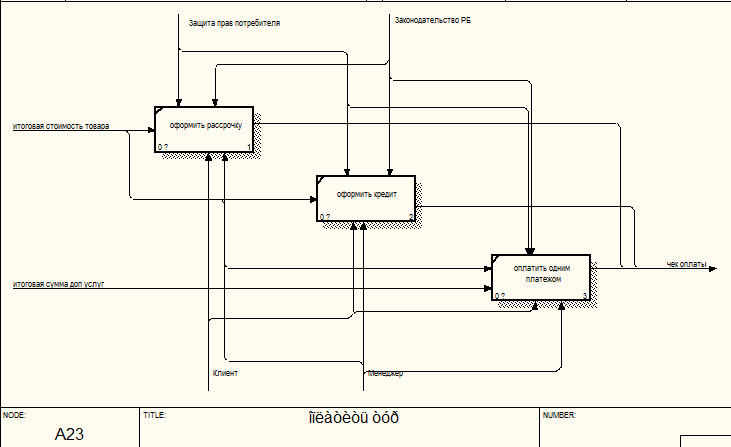


Рисунок 2.5 – Декомпозиция компонента «Оплатить тур»

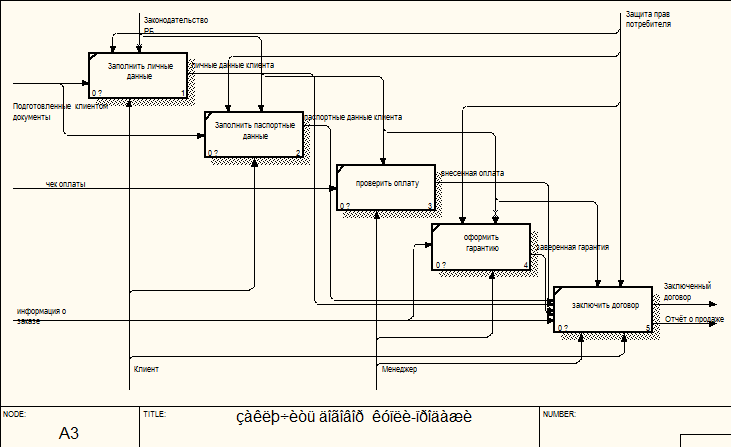


Рисунок 2.6 – Декомпозиция компонента «Заключить договор купли-продажи»

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

На основе анализа созданной модели «как-есть» (TO-BE) определим требуемые изменения процессов. В данном случае целесообразно будет ввести систему онлайн-сервиса.

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».

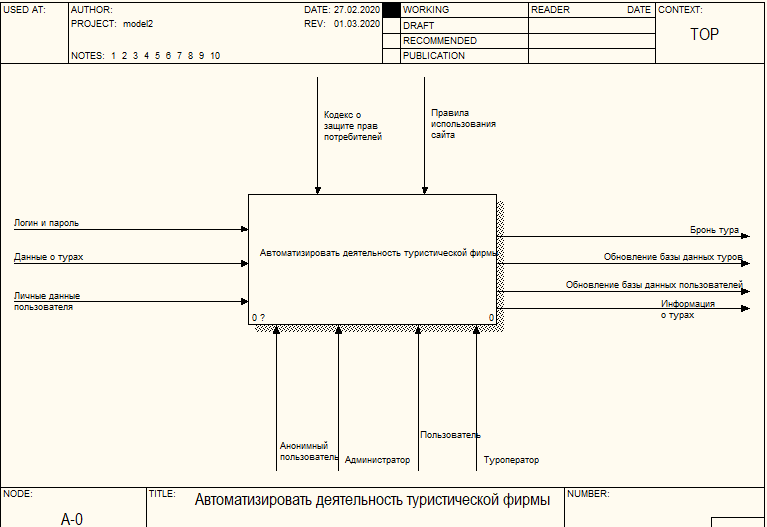


Рисунок 2.7 – Контекстная диаграмма автоматизации деятельности туристической фирмы

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».

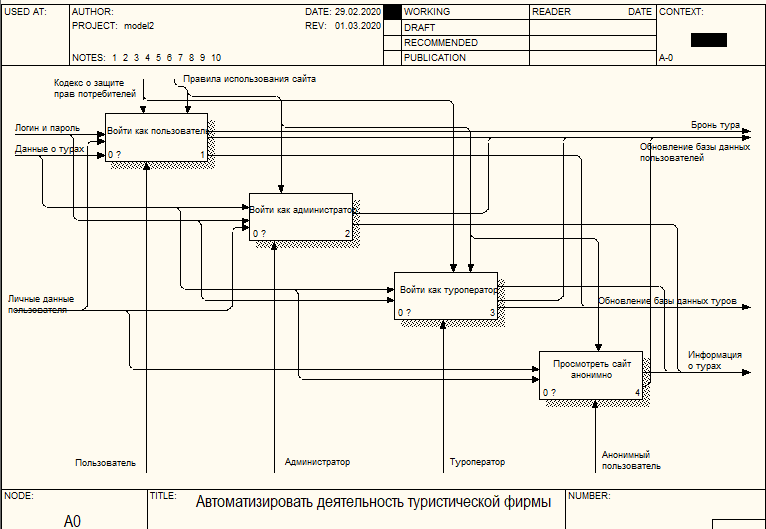


Рисунок 2.8 – Декомпозиция верхнего уровня контекстной диаграммы

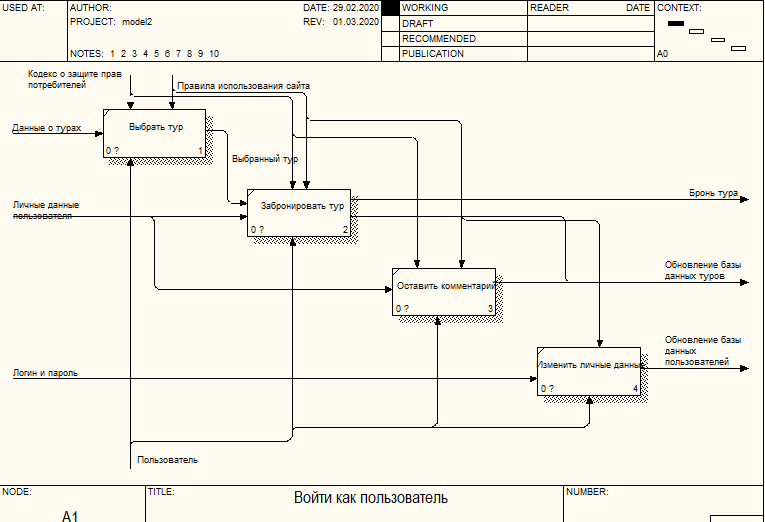


Рисунок 2.9 – Декомпозиция блока «Войти как пользователь»

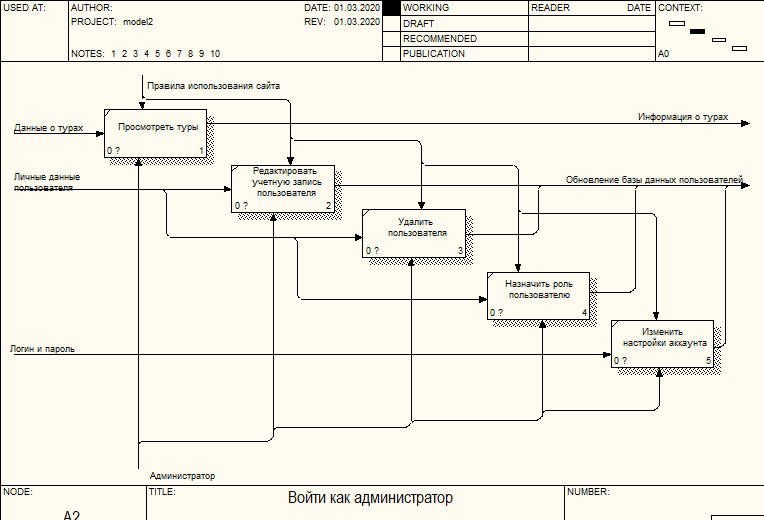


Рисунок 2.10 – Декомпозиция блока «Войти как администратор»

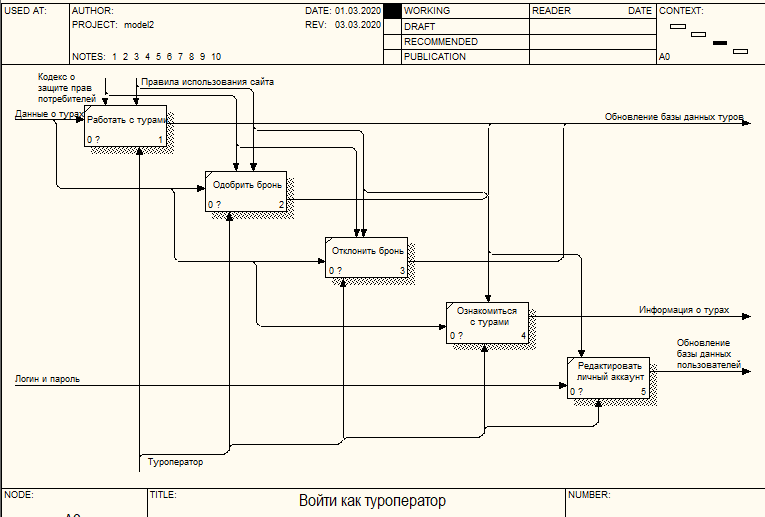


Рисунок 2.11 – Декомпозиция блока «Войти как туроператор»

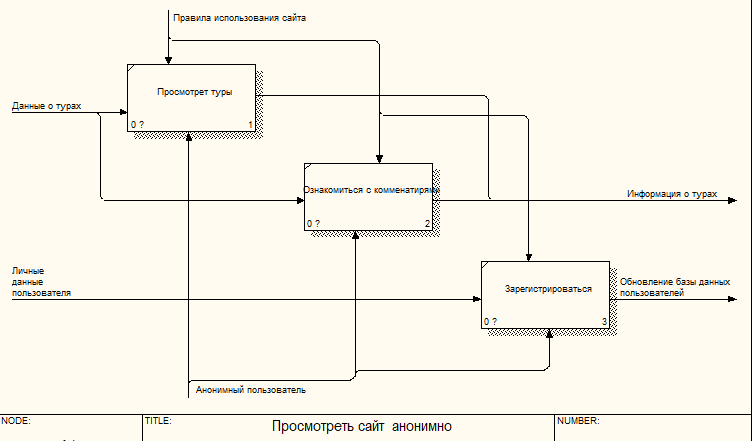


Рисунок 2.11 – Декомпозиция блока «Просмотреть сайт анонимно»

По итогу любой посетитель сайта получит то, что хочет.

# **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).

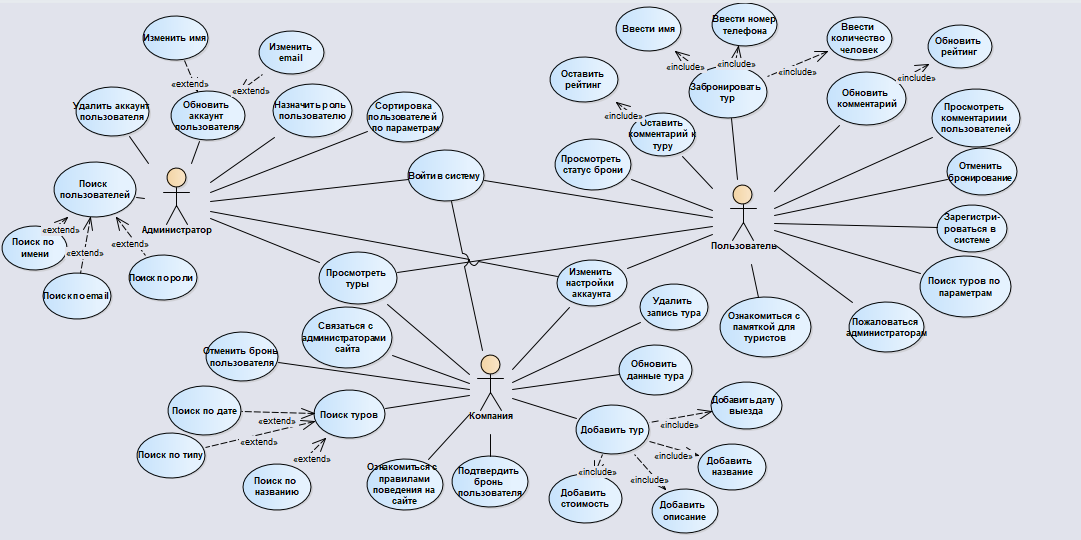


Рисунок 3.1 – Диаграмма вариантов использования

В данном проекте используются три действующих лица: пользователь, администратор и компания. Каждое лицо может выполнять определенные действия и в некоторых ситуациях действия могут повторятся, например, «Просмотреть туры», «Войти в систему» и «Изменить настройки аккаунта».

# **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)

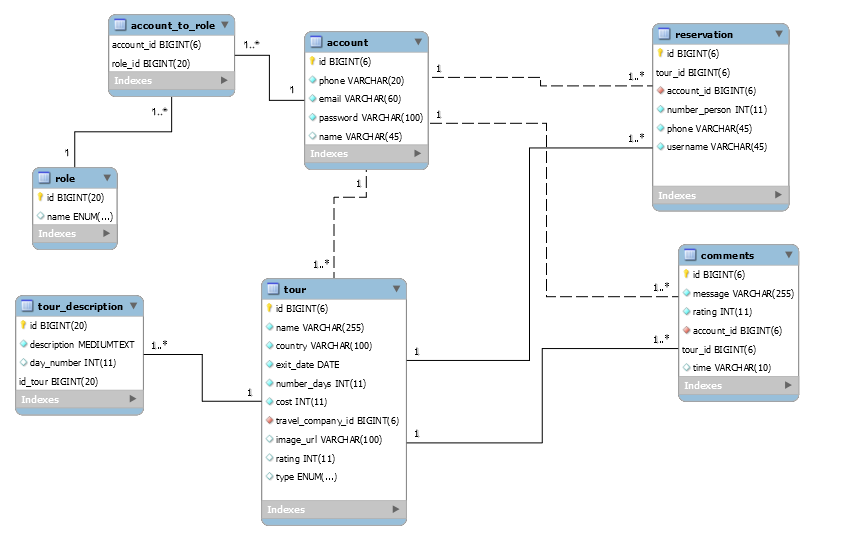


Рисунок 4.1 – Информационная модель web-приложения для туристической сферы

Для хранения данных в БД было создано 7 связанных сущностей:

* account;
* role;
* account\_to\_role;
* tour;
* tour\_description;
* reservation;
* comments.

Таблица account используется для хранения данных пользователей. Данная таблица содержит поля: id BIGINT(6), phone VARCHAR(20), email VARCHAR(60), password VARCHAR(100), name VARCHAR(45). Поля phone, email являются уникальными, id – первичный ключ.

Таблица role используется для хранении названия роли и имеет поля id BIGINT(20), name ENUM('ROLE\_ADMIN', 'ROLE\_USER', 'ROLE\_COMPANY').

Таблицы account и role связаны между собой связью многие-ко-многим через таблицу account\_to\_role, которая содержит в себе внешние ключи account\_id и role\_id. Данная таблица используется для хранения данных о ролях, которые присваиваются пользователю.

Таблица tour используется для хранения данных о турах и имеет следующие поля: id BIGINT(6), name VARCHAR(225), country VARCHAR(100), exit\_date DATE, number\_days INT(11), cost INT(11), travel\_compony\_id BIGINT(6), image\_url VARCHAR(100), rating INT(11), type ENUM('BUS\_TOUR', 'AIR\_TOUR', 'SEA\_TOUR').

Поле name содержит данные о названии тура, country – о стране назначения, exit\_date – дата выезда, number\_days – продолжительность тура, type – тип тура, который может быть автобусный, с авиаперелетом или морской. Поле travel\_company\_id является внешним ключом и связывает таблицу account и tour связью один-ко-многим.

Каждый тур должен содержит программу, которая расписана по дням. Данная программа будет содержаться в таблице tour\_description, и связана с таблицей tour связью один-ко-многим. Таблица tour\_description содержит следующие поля: id BIGINT(20), description MEDIUMTEXT, day\_number(11), id\_tour BIGINT(20) (внешний ключ).

Таблица reservation содержит информацию о бронях туров. Поля tour\_id и account\_id являются внешними ключами, которые связывают таблицу reservation с таблицами account и tour связью один-ко-многим. Также таблица содержит поля number\_person INT (количество человек), phone VARCHAR(45),username VARCHAR(45) и status ENUM('IN\_PROCESS', 'DENIED', 'APPROVED'), которое содержит информации о статусе брони (рассматривается, принята или отклонена).

Таблица comments используется для хранения комментариев, которые оставили пользователи. Таблица содержит поля: id BIGINT(11), message VARCHR(255), rating INT, account\_id BIGINT, tour\_id BIGINT, time VARCHAR(10). Поля account\_id и tour\_id являются внешними ключами и связывают таблицу comments с таблицами account и tour связью одни-ко-многим.

# **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.

Spring Core предоставляет базовые средства для создания приложений— управление компонентами (бинами, beans), внедрение зависимостей, MVC фреймворк, транзакции, базовый доступ к БД.

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.

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 .

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.

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. Чаще всего и удобнее всего использовать внешний файл, который при помощи специального тега подключается к html-документу. Такие файлы имеют расширение \*.css.

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" \o "Типографика), веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "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" \o "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.

## **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.

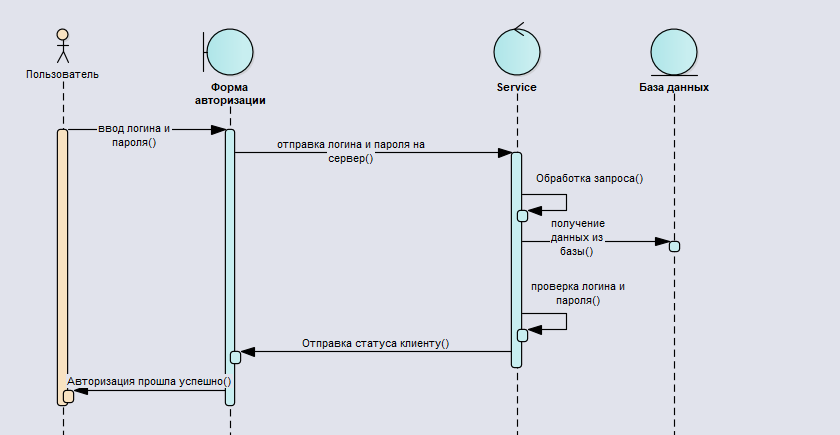


Рисунок 5.1 – Диаграмма последовательности для процесса авторизации пользователя

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 А.2 представлена диаграмма компонентов, которая обуславливает архитектуру разрабатываемой системы.

## **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 А.3).

