

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук
Кафедра программирования и информационных технологий

Система покупки электронных авиабилетов

09.03.02 Информационные системы и технологии

Курсовой проект по дисциплине «Технологии программирования»

Обучающийся _____ В. М. Беспалов, 3 курс

Обучающийся _____ Д.Р. Ступак, 3 курс

Обучающийся _____ И.П. Ткаченко, 3 курс

Руководитель _____ В.С. Тарасов

Содержание

Содержание	2
Введение	3
1 Постановка задачи	7
2 Анализ предметной области	8
2.1 Анализ существующих решений	8
2.2 Продуктовые воронки	10
2.3 Анализ средств реализации	11
3 Диаграммы, описывающие работу системы	12
3.1 Диаграмма прецедентов	13
3.2 Диаграмма классов	14
3.3 Диаграмма объектов	15
3.4 Диаграмма состояний	15
3.5 Диаграмма последовательностей	17
3.6 Диаграмма взаимодействия	21
3.7 Диаграмма активности	23
3.8 Диаграмма развёртывания	24
3.9 Диаграмма IDEF-0	25
Заключение	25
Список использованных источников	26

Введение

Актуальность работы:

Перед любой информационной системой стоит цель удовлетворить потребности людей. Потребность в транспорте и передвижении – входит в их число. Экономия времени, собственных сил и красивый вид из иллюминатора – это всё относится к авиаперелётам. Зачастую не существует иного способа, как преодоление расстояния воздушным путём. «Постоянное наращивание и модернизация объектов инфраструктуры аэропортового комплекса позволит к 2030 г. увеличить пропускную способность до 64 млн. пасс» [1]

Цифры из статистики говорят о том, что воздушный транспорт еще проходит этап становления, и в будущем он только увеличит своё влияние. Поэтому данная сфера актуальна для исследования и реализации программного продукта.

Электронный формат покупки билетов упрощает не только взаимодействие между людьми, но и даёт покупателю актуальную и структурированную информацию. Покупатель анализирует представленные данные и выбирает лучший для себя вариант.

Разработка мобильной версии системы для покупки электронных авиабилетов обусловлена следующими причинами:

- открытая статистика компании Top Annie (Платформа, анализирующая мобильные данные), говорит о том, что рынок мобильных приложений находит все больше отклика от общества. Начиная с 2019 года пользователи со всего мира провели в мобильных приложениях на 50% больше времени от 2018 года
- по оценке Criteo, (Ведущая рекламная платформа для открытого интернета): через мобильные приложения пользователи за сеанс просматривают в 4 раза больше товарных позиций, чем через адаптивный сайт. Крупные компании не задумываясь

обеспечивают свои продукты мобильными приложениями, упрощающими покупку их услуг

- наличие аналитики поведения покупателей даёт разработчикам приложений точную информацию по улучшению собственного продукта
- постоянный доступ к смартфону позволяет покупателю, не отвлекаясь от важных дел найти интересующую его услугу. Это просто, удобно и дает пользователям свободу действий

Поэтому мобильная разработка очень актуальна.

На территории России – основным потребителем мобильных приложений являются пользователи платформы Android, а именно «73% населения предпочитает Android и только 26% пользуется гаджетами на IOS» [2]. Исходя из данных сведений был сделан выбор в пользу разработки приложения для платформы Android.

Цель:

Основными целями разработки данной системы являются:

- упрощение взаимодействия покупателя с продавцом в лице авиакомпании
- предоставление актуальной информации о билетах
- повышение уровня осведомленности пользователя о ценах на авиабилеты

Задачи:

- разработать мобильное приложение для платформы Android
- провести аналитический сбор данных об активности пользователей в приложении
- создать рекомендательную систему, которая содержит рейсы, интересные пользователю

Анализ источников:

Создание выбранной системы включает в себя не только изучение документации Фреймворков, но и информации, дающей представление об авиаперевозках. С какими данными работает данная сфера, какие факторы влияют на выбор определенного билета и главное направление – каким образом преподнести пользователю информацию, чтобы он пришёл к ожидаемому результату. Исходя из этого, были изучены следующие источники:

- мобильные приложения по продаже авиабилетов (Aviasales, Аэрофлот, Туту.ру, Авиабилет)
- сервисы, предоставляющие данные о полётах (Skyscanner, Amadeus travel)
- сайты авиакомпаний с информацией, описывающей этапы покупки билетов и данные, необходимые для покупки

В ходе анализа мобильных приложений по продаже билетов, были выявлены основные виды фильтров, ускоряющие выбор покупателем билета. Также были выявлены небольшие изъяны, реализацию которых учитывала бы данная разрабатываемая система.

Проанализировав сервисы, предоставляющие данные, было уменьшено количество фильтров, доступных к разработке. Приведённая на сервисах информация не обеспечивала фильтры необходимыми сведениями.

Анализ Сайтов авиакомпаний дал представление о процессе взаимодействия покупателя с авиакомпанией, начиная от покупки билета и заканчивая регистрацией покупателя (уже пассажира) на рейс.

Проведя анализ изложенных выше источников, было разработано техническое задание, описывающее основной функционал,

интерфейс и этапы разработки мобильного приложения, а также была предложена рекомендательная схема подбора билетов.

Методология:

В качестве методологии разработки данной системы была выбрана каскадная модель разработки продуктов. Разработка системы ведётся поэтапно: каждый этап имеет ограниченные сроки, включает в себя последовательное выполнение процессов, распределение задач и ведение отчетности.

1 Постановка задачи

Составить техническое задание проекта, а именно:

- проанализировать уже существующие дизайны систем поиска и покупки товаров
- составить сценарии работы приложения
- создать дизайн конечного приложения

Разработать серверное приложение, а именно:

- создать проект приложения
- разработать схему базы данных для хранения информации
- создать сервис поиска билетов для заполнения базы данных приложения актуальной информацией
- создать сервис для взаимодействия с клиентским приложением, который будет предоставлять запрошенные данные по авиабилетам
- создать тесты для проверки работоспособности приложения перед его окончательной сборкой и развёрткой

Разработать клиентское приложение, а именно:

- создать проект приложения
- создать окна мобильного приложения с разработанным дизайном
- создать контроллеры окон приложения
- создать сервис для соединения с серверным приложением
- создать тесты для проверки работоспособности приложения перед его окончательной сборкой

В дополнение к всем вышеперечисленным пунктам необходимо:

- создать систему CI/CD для автоматизации тестирования и обновления приложений
- создать сервис для работы с Telegram для рекомендаций пользователям интересных предложений

2 Анализ предметной области

2.1 Анализ существующих решений

Анализ существующих решений будем проводить на основе данных сервисов:

Таблица 1 - Примеры существующих решений

https://bit.ly/3hktFDB	Aviasales
https://bit.ly/3bagPUt	Yandex.Flights

1. Aviasales

Общее описание:

Aviasales является метапоисковым сервисом и не продаёт билеты. Найдя нужный билет, пользователь переходит на страницу авиакассы или авиакомпании, чтобы совершить оплату. Оплата происходит на странице авиакомпании или авиакассы либо прямо в интерфейсе поисковика. В интерфейсе поисковика, то есть в формате экспресс-бронирования (assisted booking), продаются билеты «Уральских авиалиний» и S7 Airlines.

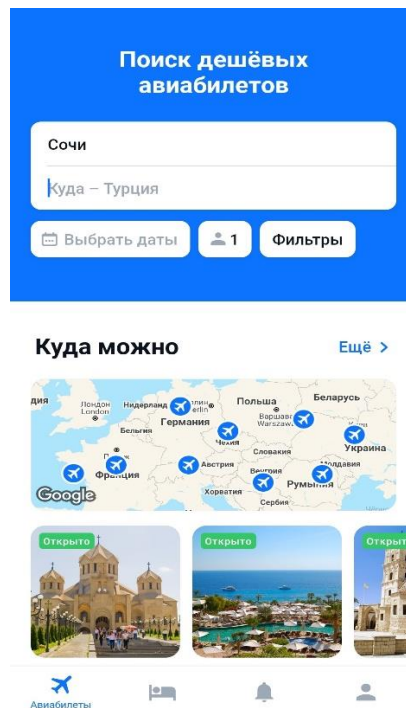


Рисунок 1 - Страница поиска Aviasales

Достоинства:

- понятный дизайн
- агрегация информация от многих перевозчиков
- большое количество страниц
- возможность забронировать отель

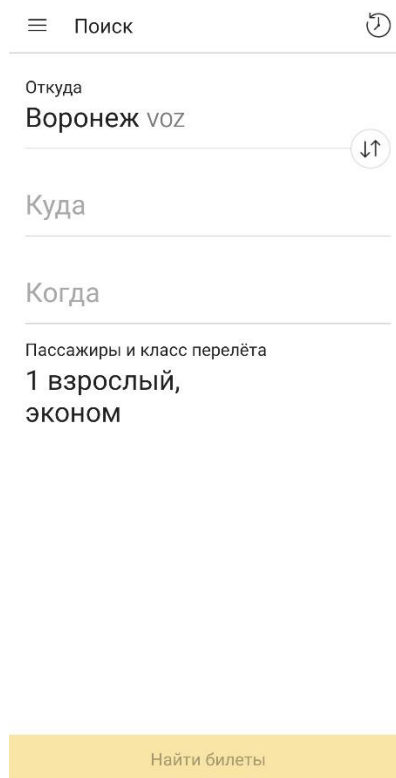
Недостатки:

- дизайн перегружен анимациями

2. Yandex.Flights

Общее описание:

Сервис для подбора авиабилетов от Яндекса. Также как и Aviasales самостоятельно не продает билеты. При покупке билетов переходит на страницу авиакомпании для покупки билетов.



Поиск

Откуда
Воронеж VOZ

Куда

Когда

Пассажиры и класс перелёта
1 взрослый,
эконом

Найти билеты

Рисунок 2 - Страница поиска Yandex.Flights

Достоинства:

— Простой интерфейс поиска

Недостатки:

— Низкая скорость работы

— Большое количество рекламы

2.2 Продуктовые воронки

Рассмотрим количество шагов, которые необходимо пройти до основных функций приложения – покупка билета. При входе, пользователю необходимо ввести пункты назначения, а также дату. Далее, ему предлагается список рейсов на выбор. При нажатии на рейс, открывается окно с выбором рейса, где возможно приобрести билет. Таким образом, всего за четыре экрана возможно приобрести

билет.

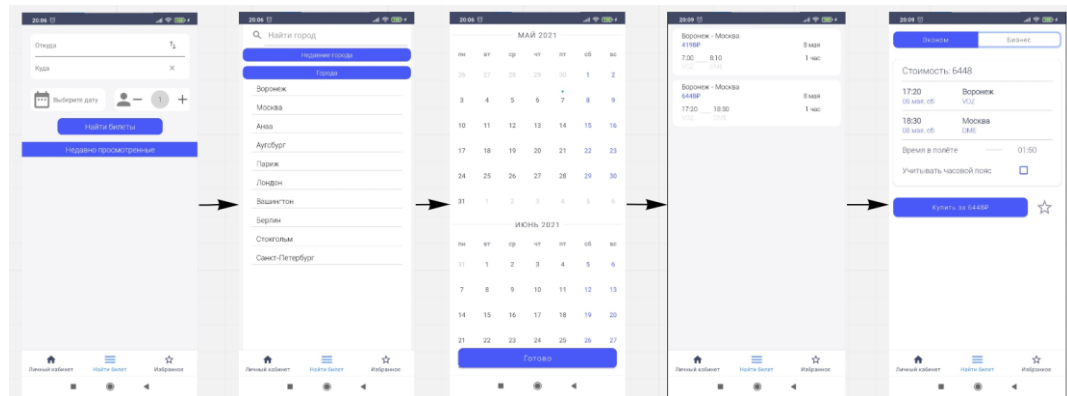


Рисунок 3 - Продуктовая воронка

2.3 Анализ средств реализации

В качестве средств реализации системы поиска авиабилетов были выбраны следующие технологии:

1. PostgreSQL была выбрана в качестве СУБД. Она является бесплатной наиболее популярной в данный момент и регулярной обновляемой системой управления с регулярно обновляемой базой данных.
2. В качестве IDE была выбрана среда AndroidStudio из-за того, что именно она рекомендуется для разработки Android-приложений под большинство версий данной системы. Она позволяет удобно проектировать приложения и проверять работу кода.
3. IntelliJ IDEA была выбрана в качестве среды разработки для API. Она выбрана из-за удобства использования в процессе работы. В процессе написания кода программистом она занимается построением синтаксического дерева, определением особенностей размещенных ссылок, анализом возможных путей исполнения операторов и передачи данных. Основываясь на полученных результатах, программа обращает внимание специалиста на существующие ошибки и самостоятельно устраняет их, предоставляет варианты автоматического

дополнения кода. Благодаря указанным особенностям она избавляет пользователя от повседневной рутины и позволяет ему сконцентрироваться на более важных задачах.

4. В качестве языка разработки используются Java, так как данный язык является широко распространенным в современном мире вследствие своей универсальности. Именно на данном языке написан гибкий и удобный фреймворк Spring, при помощи которого реализован API приложения.
5. В качестве сервера для работы с API используется Heroku из-за простоты использования и стабильности работы. Также Heroku позволяет использовать Docker контейнеры для запуска своего backend-приложения.

3 Диаграммы, описывающие работу системы

Ввиду сложности разработки системы, были разработаны UML диаграммы, которые, ввиду универсальности, должны помочь проверяющему понять принципы работы системы.

3.1 Диаграмма прецедентов



Рисунок 4 - Диаграмма прецедентов

Каждый пользователь при взаимодействии с приложением имеет ряд возможностей, который наглядно представлен на рис. 4:

Авторизованный пользователь может:

- Искать рейсы
- Просматривать детальную информацию о рейсе
- Добавлять рейс в избранные
- Удалять рейс из избранных
- Покупать билеты
- Просматривать историю платежей
- Выходить из аккаунта

Неавторизованный пользователь может:

- Авторизоваться
- Искать рейсы
- Просматривать детальную информацию о рейсе

Администратор может:

- Просматривать статистику о использовании приложения другими пользователями

3.2 Диаграмма классов

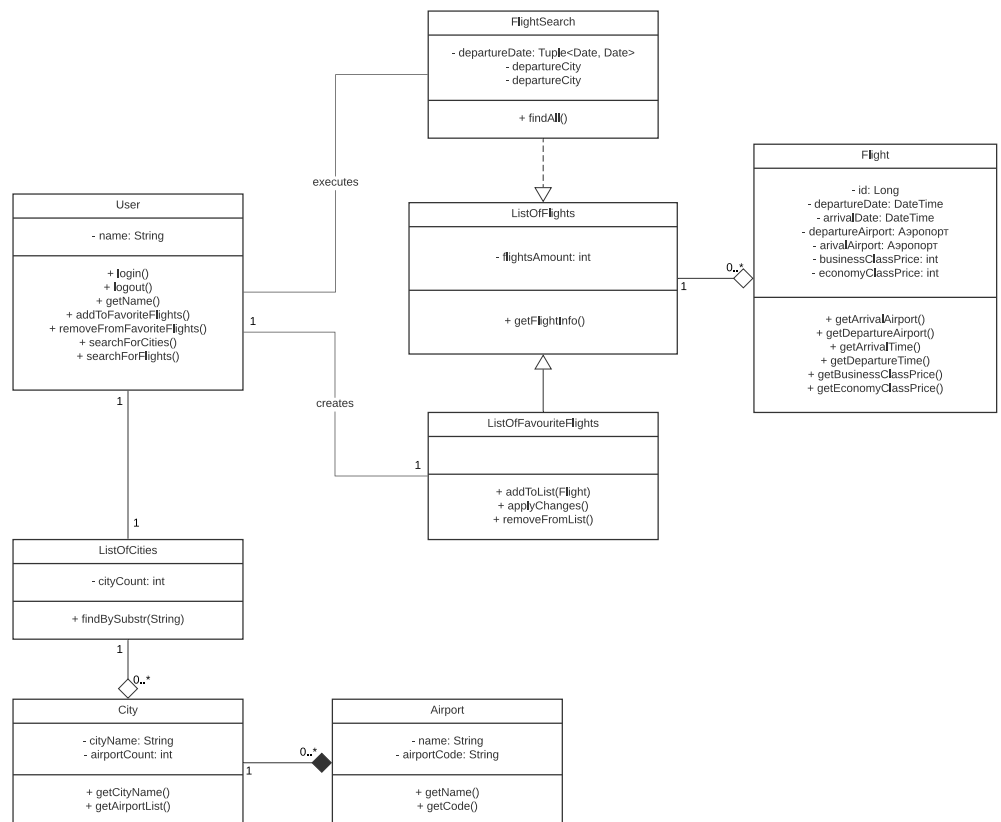


Рисунок 5 - Диаграмма классов

На рисунке 5 изображена диаграмма классов, отображающая основные компоненты системы и их взаимодействие.

3.3 Диаграмма объектов

Диаграмма объектов, изображенная на рисунке 6, отображает экземпляры классов и взаимодействие между ними.

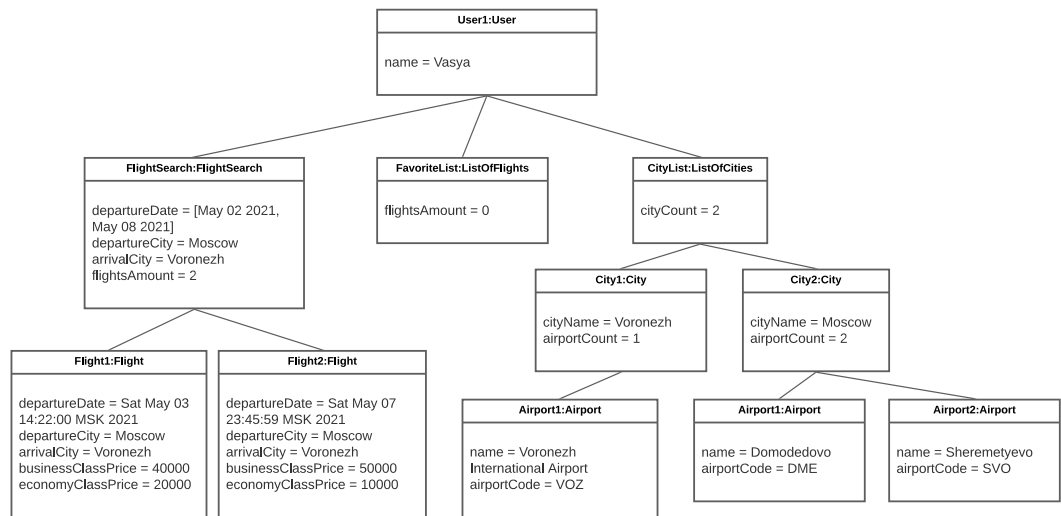


Рисунок 6 - Диаграмма объектов

3.4 Диаграмма состояний

Диаграмма состояний описывает состояния основных объектов системы.

На диаграмме, представленной на рисунке 6 представлено изменение состояний объекта пользователь. Начальное состояние представлено неавторизованным пользователем. Авторизация пользователя происходит с участием дополнительного сервиса. После успешной авторизации пользователь переходит в свое конечное состояние – авторизованный пользователь. В случае неуспешного входа будет предложена попытка повторного входа.



Рисунок 7 - Авторизация пользователя

Следующая диаграмма, отображенная на рисунке 8, описывает изменение состояний объекта рейс.

Начальное состояние включает в себя экран для поиска рейсов. После выбора фильтров для рейса начальное состояние переходит в состояние «Рейс с фильтрами». Далее следует поиск рейсов. В случае если рейсы не найдены, то состояние рейса не изменится, иначе состояние перейдет в состояние «Найденные рейсы». Далее представлена альтернатива – оформление покупки билета и добавление в избранное. Оба процесса требуют авторизации пользователя. При оформлении покупки - помимо выбора рейса, требуется информация о пассажирах. В этом случае рейс меняет состояние на «Билет на рейс с данными о пассажирах», а после оплаты – «Купленные билеты на рейс». В случае добавления рейса в избранное – состояние изменяется на «Избранный рейс». Сущность приходит в свое конечное состояние.

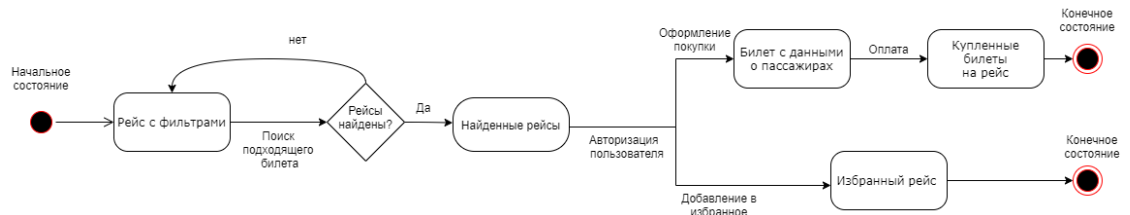


Рисунок 8 - Диаграмма состояний рейса

Диаграмма, отображенная на рисунке 9, описывает изменение состояний для сущности администратор.

Начальное состояние представлено пользователем сайта. После прохождения авторизации начальное состояние переходит в состояние авторизованный администратор, которому доступна функция: получение статистики. Сущность перешла в свое конечное состояние.



Рисунок 9 - Диаграмма состояния администратора

3.5 Диаграмма последовательностей

Диаграмма последовательностей включает в себя временное описание процессов, протекающих в системе. По горизонтали указаны объекты, участвующие в процессе, по вертикали – промежутки активности объектов.

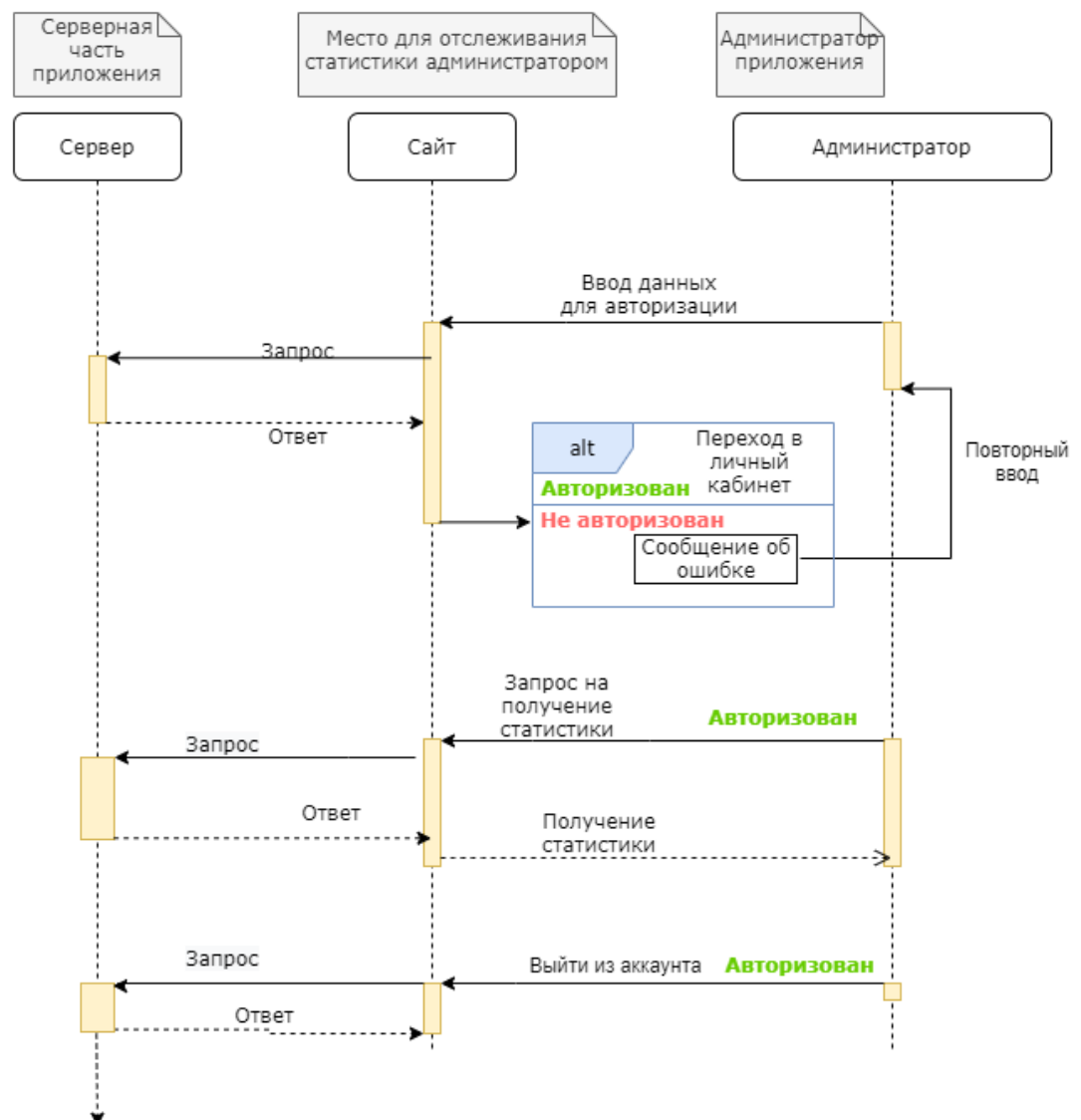


Рисунок 10 - Последовательности действий администратора

На диаграмме, отображенной на рисунке 10, представлено взаимодействие объектов – сервер, сайт, администратор. Для возможности выполнения действий – администратору необходимо пройти авторизацию. Авторизация проходит с участием серверной части приложения. После получения ответа сервера происходит или повторный ввод данных или открывается доступ к личному кабинету. После успешной авторизации администратору доступна функция получения статистики о пользователях. При условии, что администратор авторизован – он может выйти из аккаунта.

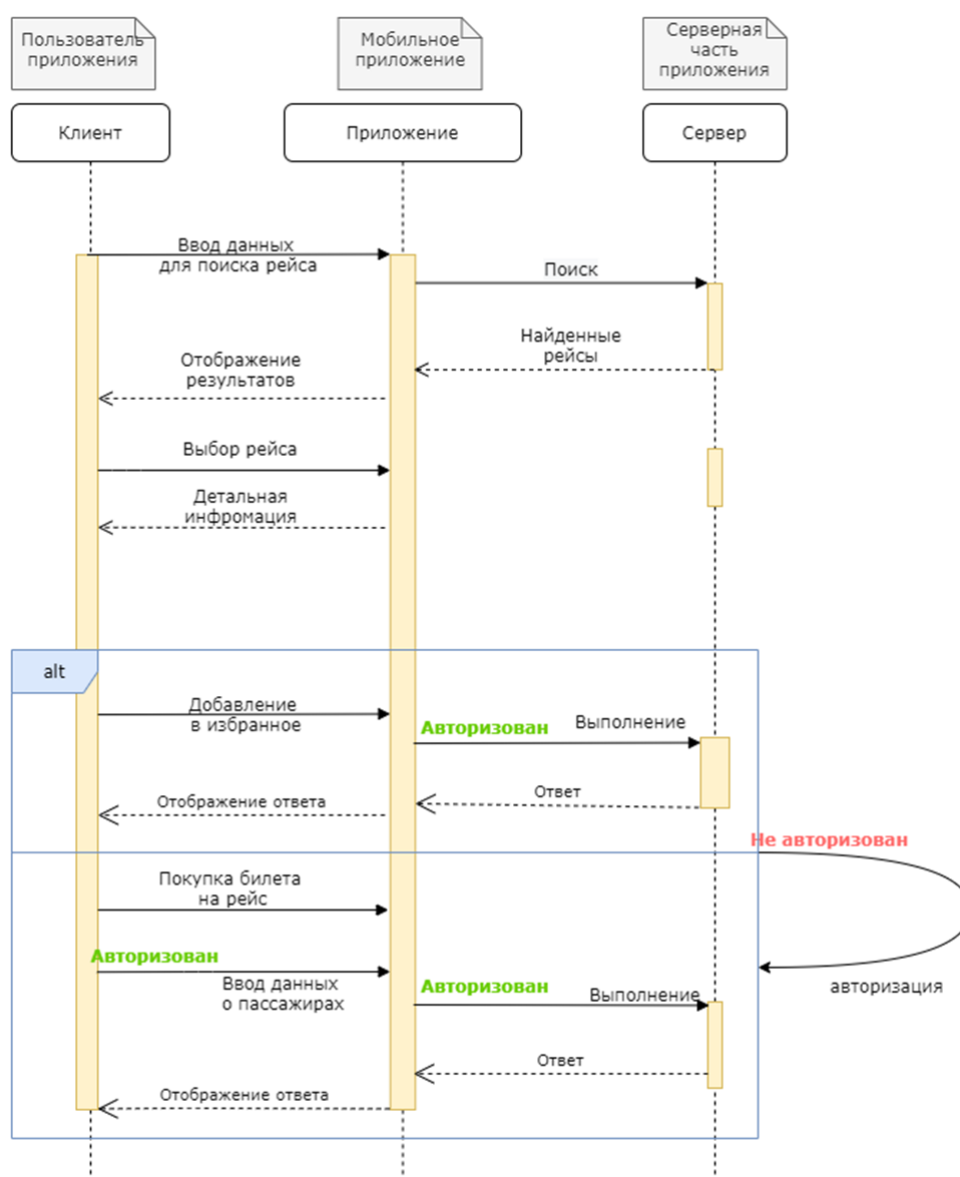


Рисунок 11 - Диаграмма последовательностей действий пользователя, сервера и приложения при работе с рейсами

Для сущности пользователь представлены операции поиск рейсов, добавление в избранное и покупка билетов.

Для поиска рейсов, пользователю необходимо ввести данные для фильтра. После запроса к серверу на экране приложения будут отображены найденные рейсы с указанными фильтрами. Теперь пользователю доступен детальный просмотр рейса и альтернативный выбор следующих действий: добавление в избранное или покупка билетов на рейс.

Для добавления в избранное пользователь должен быть авторизован. Авторизация проходит на стороне клиента с участием дополнительного сервиса. В случае, если пользователь уже авторизован, происходит запрос на сервер с выполнением операции добавления в избранное. После ответа сервера происходит отображение его ответа на экране приложения.

При покупке билетов на рейс также необходима авторизация, после прохождения авторизации необходимо заполнить сведения о пассажирах. За этим следует запрос на сервер. После ответа сервера происходит отображение его содержимого на экран приложения.

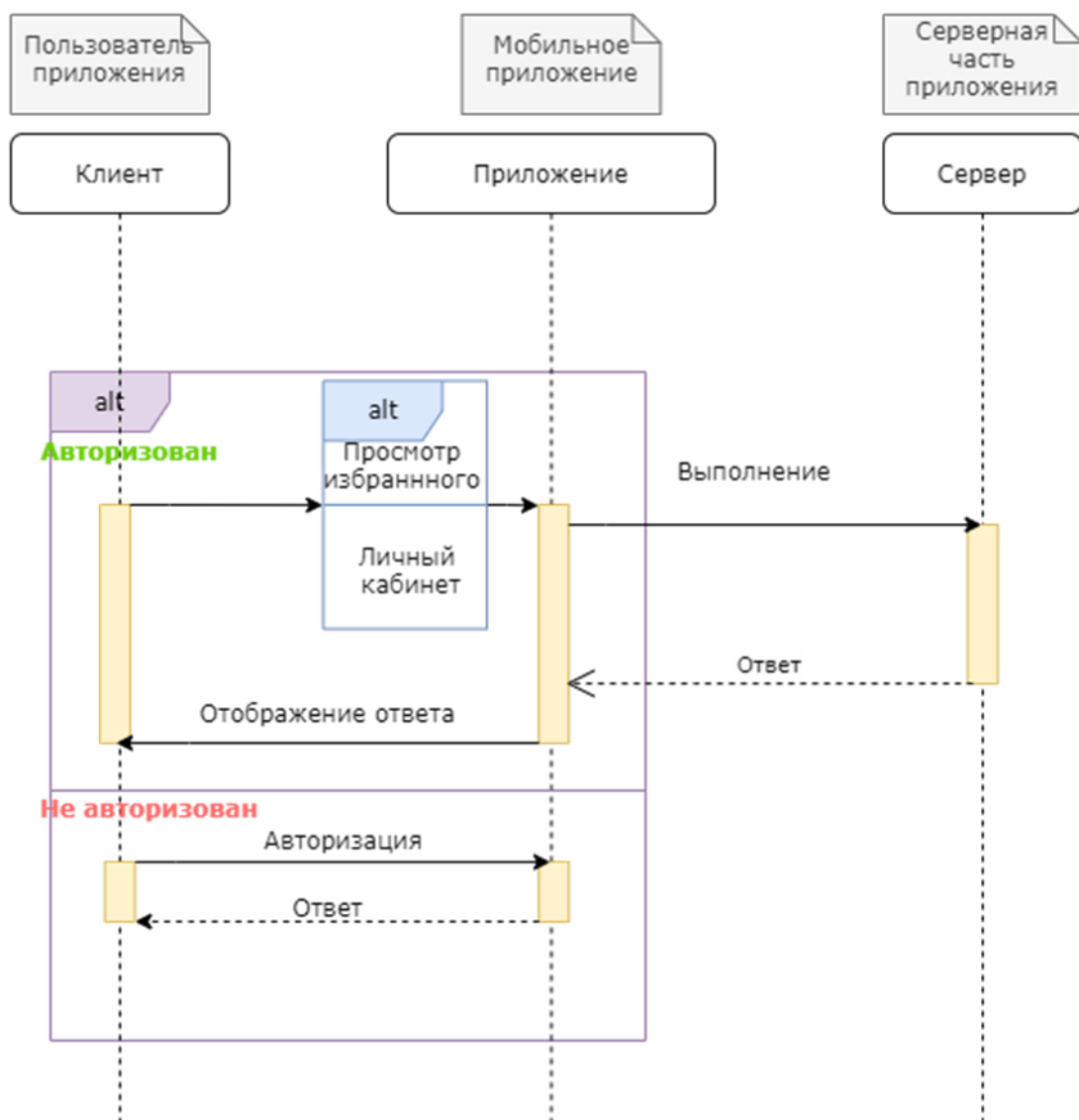


Рисунок 12 - Последовательности при работе с авторизованным или неавторизованным клиентами

В продолжение функций, доступных пользователю: операции просмотра избранного и просмотра личного кабинета. На диаграмме представлен альтернативный блок: авторизованный – неавторизованный пользователь. Если пользователь авторизован – ему доступны функции просмотра, избранного с дальнейшим удалением рейсов из него и просмотр личного кабинета. Обе функции делают запросы к серверу и отображают на экран приложения информацию, содержащуюся в ответе. В случае, если пользователь

неавторизованный - ему будет предложена авторизация с участием вспомогательного сервиса.

3.6 Диаграмма взаимодействия

Диаграмма взаимодействия описывает взаимодействия объектов системы без учета временной составляющей, лишь последовательность действий, описывающих каждую операцию.

На рисунке 13 представлены объекты пользователь, приложение, сервер. Пользователю предоставлена операция по поиску рейса с заданными фильтрами. Приложение отправляет запрос на сервер, сервер обрабатывает запрос и отправляет ответ, который будет предоставлен пользователю для дальнейшего изучения.



Рисунок 13 - Взаимодействия при поиске рейсов с заданными фильтрами

Диаграмма, изображенная на рисунке 14 описывает авторизацию пользователя с участием приложения. Приложение использует сторонний сервис для авторизации. После попытки авторизации будет получен ответ, прошел ли пользователь авторизацию.

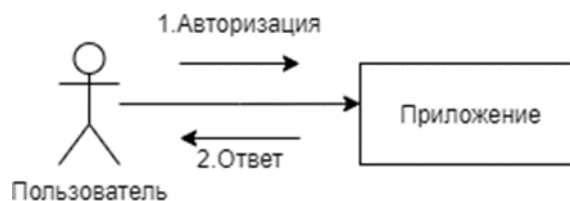


Рисунок 14 - Взаимодействия при авторизации

На рисунке 15 основное действие выполняется авторизованным пользователем. Пользователь производит покупку билетов. После внесения данных о пассажирах будет предложена оплата билетов. В случае покупки – приложение отправит соответствующий запрос на

сервер и получит ответ, который приложение отобразит пользователю.



Рисунок 15 - Взаимодействия при покупке билетов

На рисунке 16 основное действие выполняется авторизованным пользователем. Пользователь производит добавление рейса в избранное. После добавления – приложение отправит соответствующий запрос на сервер и получит ответ, который приложение отобразит пользователю.

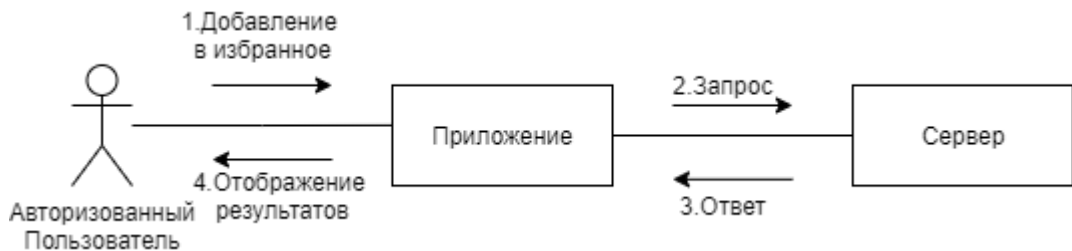


Рисунок 16 - Взаимодействия при добавлении в избранное

На диаграмме, отображенной на рисунке 17, основное действие выполняется авторизованным пользователем. Пользователю предложены операции по просмотру избранных рейсов и личного кабинета. После выбора нужной операции приложение отправит соответствующий запрос на сервер и получит ответ, который приложение отобразит пользователю.



Рисунок 17 - Взаимодействия при просмотре личного кабинета

Далее представлены диаграммы для объекта администратор.

На рисунке 18 представлена авторизация администратора. Приложение отправляет соответствующий запрос на сервер. Сервер присылает ответ приложению, а приложение сообщает администратору об успешности или неуспешности авторизации.



Рисунок 18 - Взаимодействия администратора при авторизации

На диаграмме, изображенной на рисунке 19, представлен выбор операций, доступных авторизованному администратору. Операции: просмотр статистики и выход из аккаунта. Приложение отправляет соответствующий запрос на сервер. Сервер присылает ответ приложению, а приложение отображает данные, присланные сервером.



Рисунок 19 - Взаимодействия при просмотре статистики

3.7 Диаграмма активности

Данная диаграмма, отображенная на рисунке 20, показывает, какие действия будут выполняться в системе.

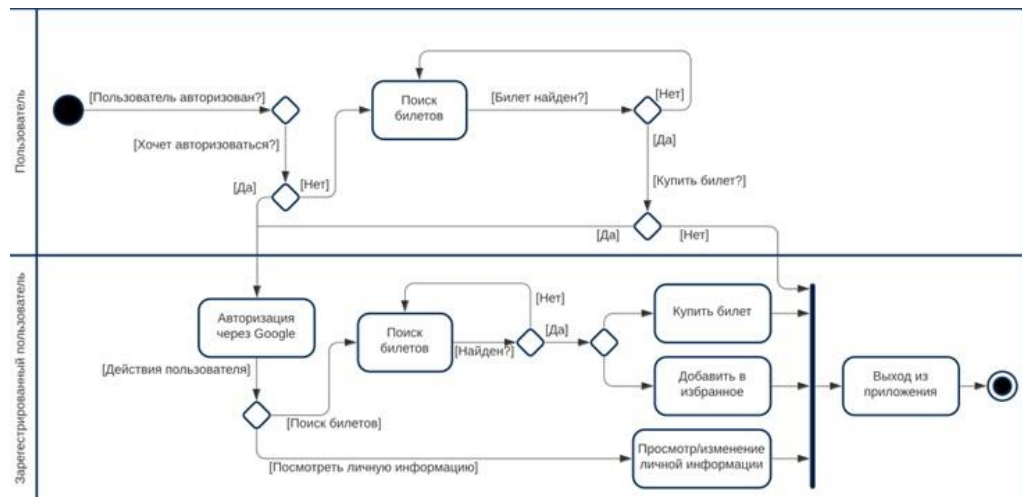


Рисунок 20 - Диаграмма активности

На данной диаграмме представлено два типа пользователей: неавторизованные и авторизованные.

Неавторизованным пользователям доступны только активности поиска билетов и авторизации.

Авторизованные пользователи могут искать билеты, покупать их или добавлять в избранное и Просматривать свою личную информацию (список избранных билетов, историю покупок).

3.8 Диаграмма развёртывания

Данная диаграмма показывает, где и какие части приложения расположены физически и как они связаны.

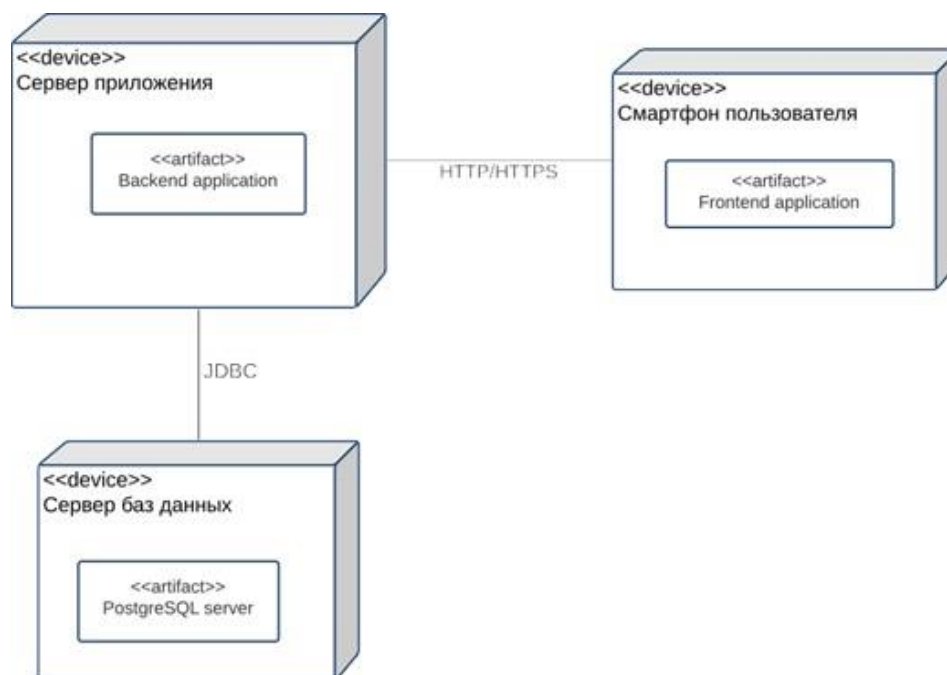


Рисунок 21 - Диаграмма развёртывания

На данной диаграмме видно, что приложение состоит из трёх частей, расположенных на трёх девайсах: серверная часть приложения, база данных и клиентская часть приложения.

Серверная часть связывается с базой данных посредством драйвера JDBC и интернет соединения.

Клиентская часть приложения связывается с серверной посредством HTTP/HTTPS запросов.

3.9 Диаграмма IDEF-0

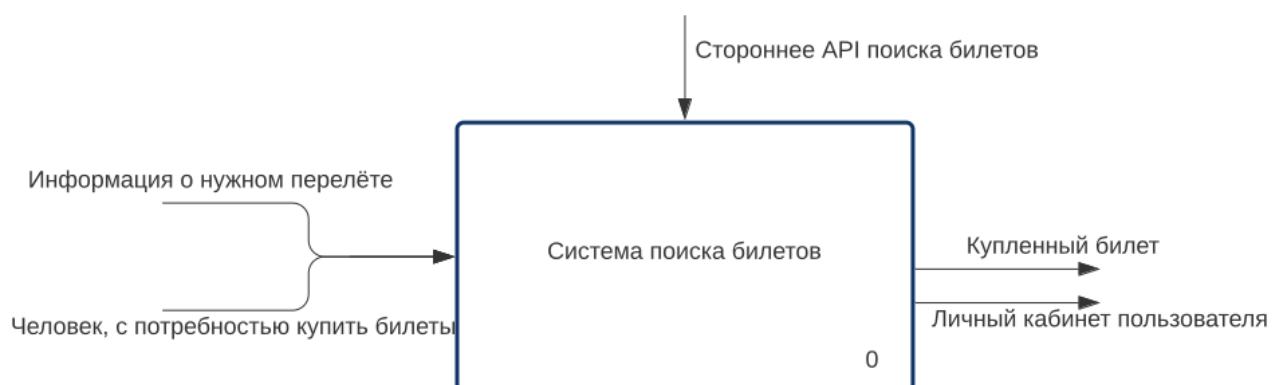


Рисунок 22 - Диаграмма IDEF-0

Заключение

Список использованных источников

1. Филина, В.Н. Стратегии развития рынка авиационных транспортных услуг / В.Н. Филина; Проблемы прогнозирования, 2019
2. Bloomberg URL: <https://kanobu.ru/news/kto-populyarnее-vmire-ios-ili-android-novaya-detalnaya-infografika-otvechaet-naetot-izvechnyj-vopro-417479/> (дата обращения: 23.03.2021).

Приложения