

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук
Кафедра информационных систем в телекоммуникациях

Система покупки электронных авиабилетов «Aviatickets»

Курсовой проект

09.03.02 Информационные системы и технологии
Информационные системы в телекоммуникациях

Руководитель_____В.С. Тарасов, ст. преподаватель____.____20__
Обучающийся_____Д.А. Загреба, 3 курс, д/о
Обучающийся_____Н.А. Ягодинцев, 3 курс, д/о
Обучающийся_____К.Р. Ходжаев, 3 курс, д/о
Обучающийся_____Я. Сулиман, 3 курс, д/о
Руководитель_____К.В. Зенин, ассистент

Воронеж 2023

Содержание

Введение.....	3
1 Постановка задачи	4
1.1 Требования к разрабатываемой системе	5
1.1.1 Функциональные требования.....	5
1.1.2 Требования к интерфейсу.....	5
1.1.3 Задачи, решаемые в процессе разработки	5
2 Анализ предметной области	7
2.1 Глоссарий	7
2.2 Цели создания программы.....	7
2.3 Сфера применения.....	8
2.4 Входные-выходные данные.....	8
2.5 Технический обзор	9
2.6 Сравнение аналогов.....	9
2.7 Требования к функциональности.....	11
2.8 Требования, не касающиеся функциональной части.....	11
2.8.1 Требования к программному обеспечению	11
2.8.2 Требования к архитектуре системы	11
2.8.3 Требования к используемым технологиям	12
2.8.4 Дизайн веб-приложения	12
2.9 Диаграмма активностей	13
2.10 Диаграмма последовательностей.....	13

Введение

В современном мире большая часть повседневных процессов автоматизирована. Хорошим примером автоматизации являются сервисы покупки авиабилетов. Они помогают клиентам искать авиарейсы, указав лишь желаемые даты для путешествия, при этом оставляя за собой выбор авиакомпании, основываясь на протяженности перелета, количестве пересадок и цене билетов.

Целью данной работы является разработка веб-приложения с возможностью покупки авиабилетов с фильтрацией и с API для интеграции любых авиаперевозчиков в систему.

1 Постановка задачи

Целью курсового проекта является создание веб-приложения. Основную функциональность разрабатываемого приложения отражает диаграмма прецедентов, изображенная на рисунке 1.

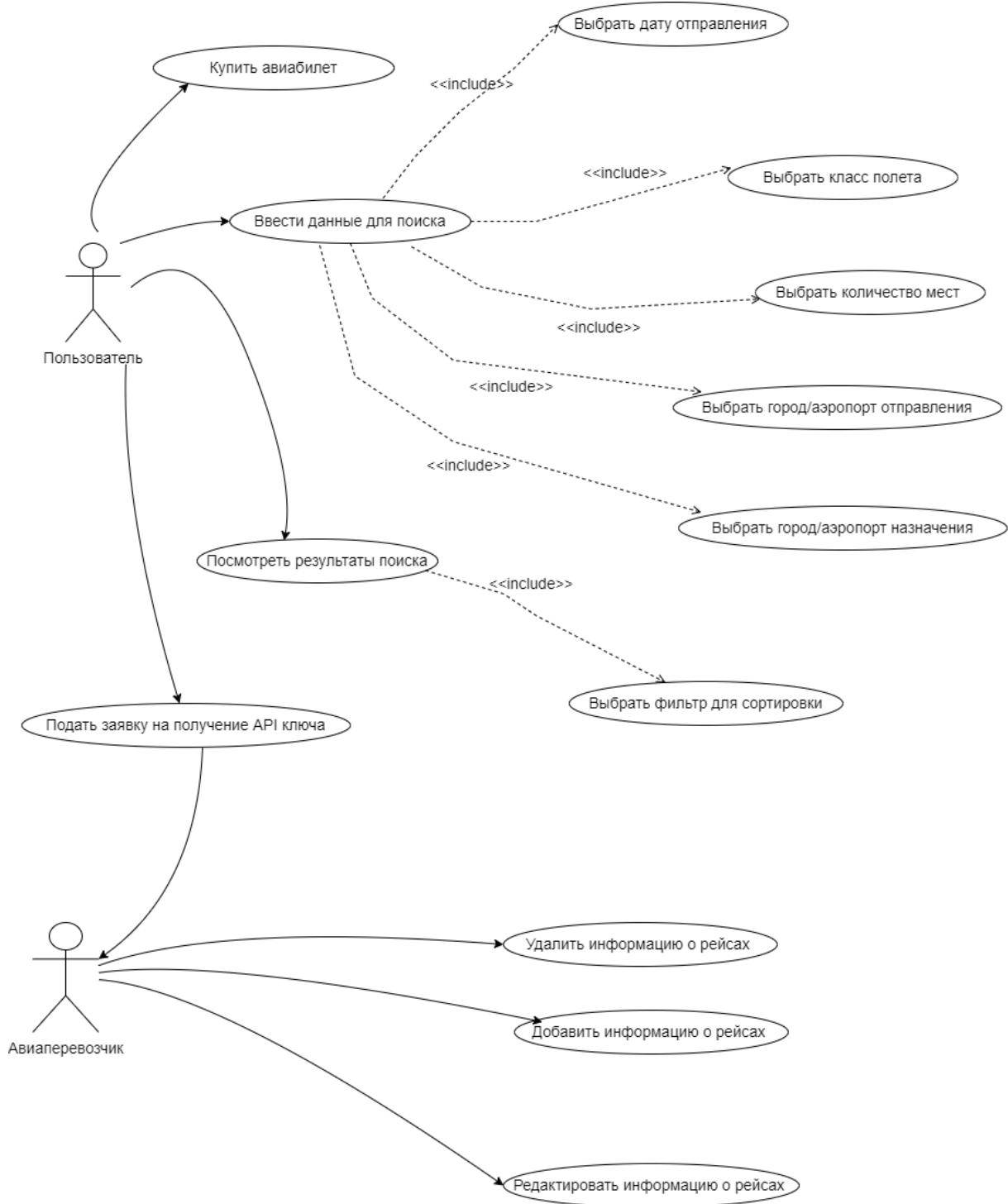


Рисунок 1 - Диаграмма прецедентов

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

- Возможность выбора откуда и куда осуществляется перелет;
- Возможность выбора даты перелета;
- Возможность фильтрации предложений;
- Для авиаперевозчиков: возможность создавать, редактировать, удалять рейсы;
- Для авиаперевозчиков: возможность изменять информацию о себе;
- Для авиаперевозчиков: возможность отправлять заявки на получение API ключа.

1.1.2 Требования к интерфейсу

- Выполнен в едином стиле и цветовой гамме;
- Текст должен быть хорошо виден на экране;
- Интерфейсы должны быть на русском языке;
- Содержать только необходимую для пользователя информацию.

1.1.3 Задачи, решаемые в процессе разработки

- Анализ предметной области;
- Написание технического задания;
- Проектирование веб-приложения;
- Разработка back-end части:
 - Реализация ролей:
 - Пользователь
 - Авиаперевозчик
 - Разработка базы данных;
 - Подключение внешнего модуля для хранения данных;
- Разработка front-end части, включающей в себя:
 - Создание макета дизайна в Figma;

- Реализация макета дизайна;
 - Интеграция API сервера;
- Проведение тестирования проекта.

Завершенный проект представляет собой полностью функционирующее веб-приложение, соответствующее требованиям, описанным выше.

2 Анализ предметной области

2.1 Глоссарий

Front-end — пользовательский интерфейс компьютера или любого устройства.

Пользователь — лицо, которое использует действующую систему для выполнения конкретной функции.

Ошибка (Bug) — общий термин, используемый для обозначения непредвиденной ошибки или дефекта в аппаратном или программном обеспечении, что приводит к его неисправности.

Веб-сервис — идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.

Сервер — обслуживающее устройство в системах обработки информации.

Метрика — термин, обозначающий любой показатель используемый в веб-аналитике для оценки эффективности какой-либо активности.

Интерфейс — набор средств, с помощью которых пользователь получает возможность взаимодействовать с системой.

REST — архитектурный стиль взаимодействия компонентов распределенного приложения в сети.

API — программный интерфейс приложения. Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

Агрегатор — приложение, объединяющее в себе услуги нескольких компаний, данные из нескольких источников и/или функции нескольких приложений или сайтов.

2.2 Цели создания программы

Система предназначена для поиска авиабилетов.

Основным назначением является поиск и покупка авиабилетов. Также предусмотрена сортировка результатов поиска по фильтрам, выбранным пользователем, и API для интеграции авиаперевозчиков.

2.3 Сфера применения

Приложение для покупки электронных авиабилетов может использоваться в различных сферах, включая личные поездки, семейные путешествия, командировочные поездки и т.д. Оно помогает выбрать авиаперевозчика, подобрать нужную цену и выбрать удобное расположение с семьей/коллегами. А также авиаперевозчики смогут разместить свои предложения на сайт.

2.4 Входные-выходные данные

Рассмотрим основной бизнес – процесс на примере IDEF0 диаграммы, представленной на рисунке 2. Данная диаграмма представляет собой общее видение процесса работы веб-приложения.

Работу сервиса регулирует законодательство Российской Федерации, расписание рейсов и нормативные документы.

На вход в систему поступает Пользователь – пассажиры без авиабилета. Обеспечивает работу веб-приложения авиаперевозчики.

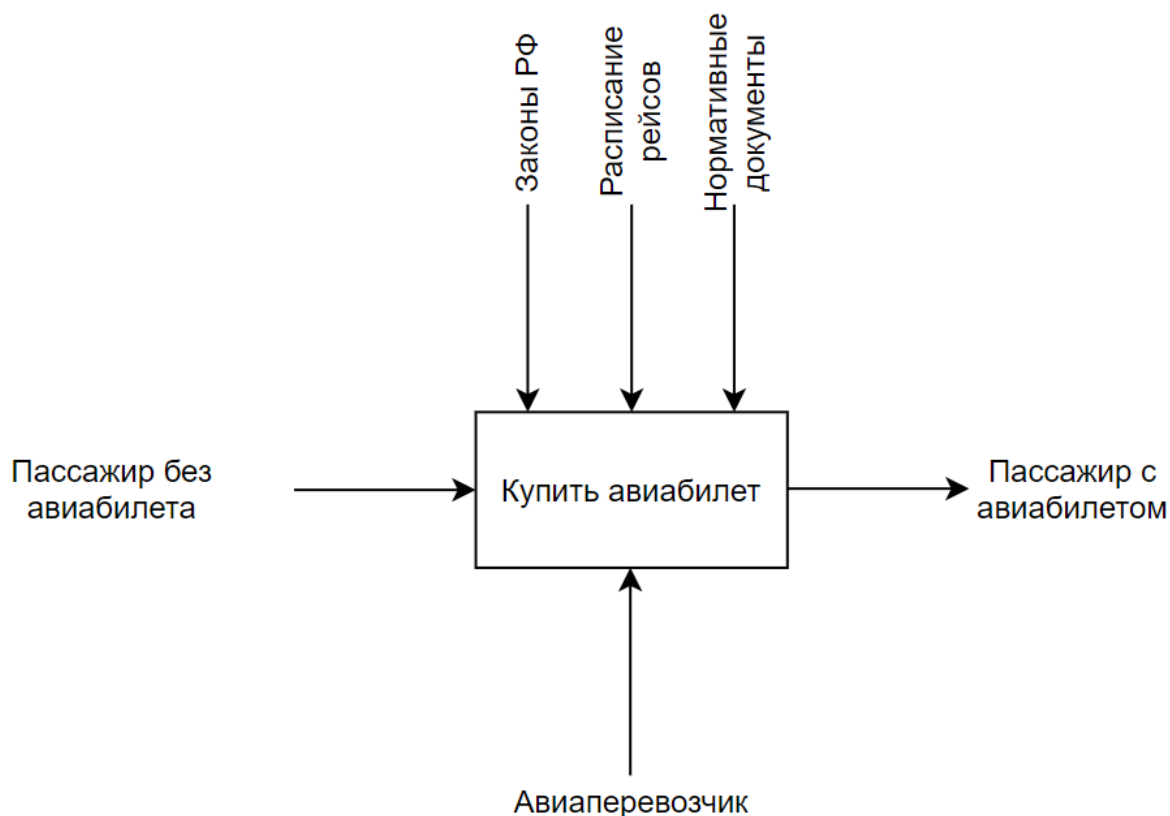


Рисунок 2 - Диаграмма IDEF0

2.5 Технический обзор

Учет интересов пользователя. Пользователь может выбрать доступного авиаперевозчика, выбрать желаемое место в самолете.

Фильтрация информации о рейсах. Пользователь вправе фильтровать по количеству билетов, сортировка по стоимости билетов, по времени вылета.

Возможность купить авиабилет.

Возможность авиаперевозчиков отправлять заявки на получение API ключа.

2.6 Сравнение аналогов

В области поиска авиабилетов существует большое количество веб-приложений. Приступая к разработке нового ПО в IT сфере, были проанализированы уже существующее ПО. Были рассмотрены их достоинства, недостатки. Насколько они удобны в использовании, содержат ли необходимый функционал. И на основе этого анализа были сделаны

выводы о том каким будет разрабатываемый продукт. А именно каким образом будет построен удобный для пользователя, интерфейс, и какой функционал будет иметь ПО.

Рассмотрим наиболее крупные из них:

Aviasales — самый популярный в России сайт для поиска авиабилетов. Это агрегатор, то есть он сам ничего не продает, а только ищет по огромному количеству источников.

Плюсы:

- Прост в использовании;
- Имеет хорошую репутацию среди пользователей и высокий уровень сервиса;
- В большинстве случаев находит билеты дешевле, чем у конкурентов.

Минусы:

- Авиабилеты не продаются;
- Aviasales плохо работает с официальными (API) сайтами авиакомпаний: предлагает в первую очередь купить билет у посредников;
- Ограниченное количество критериев для поиска и фильтрации.

Skyscanner — онлайн сервис для поиска и покупки авиабилетов.

Плюсы:

- Широкий выбор направлений для туризма;
- Хорошая выборка по цене, классу обслуживания и компании-перевозчика;
- Удобный интерфейс.

Минусы:

- Нестабильная работа;
- Авиабилеты не продаются;
- Работает лишь с ближайшим периодом;

- Цены утрачивают актуальность.

2.7 Требования к функциональности

Для пользователя должны быть предоставлены следующие функции:

- Возможность выбора откуда и куда осуществляется перелет;
- Возможность выбора даты перелета;
- Возможность фильтрации предложений;

Для авиаперевозчиков:

- возможность создавать, редактировать, удалять рейсы;
- возможность изменять информацию о себе;
- возможность отправлять заявки на получение API ключа.

2.8 Требования, не касающиеся функциональной части

2.8.1 Требования к программному обеспечению

Приложение должно удовлетворять следующим требованиям: Сайт должен работать в следующих браузерах: Google Chrome (начиная с версии 90), Firefox (начиная с версии 90) и Microsoft Edge (начиная с версии 90);

Должно предоставлять авиаперевозчикам техническую возможность интеграции с программным обеспечением и размещать свои билеты на продажу.

2.8.2 Требования к архитектуре системы

Сайт должен иметь архитектуру, соответствующую модели клиент-серверного взаимодействия.

Серверная архитектура должна быть реализована в соответствии с паттерном MVC. С четким разделением на model (модель), view (вид), и controller (контроллер).

Пользователи взаимодействуют с сайтом с помощью клиентской части веб-приложения. Взаимодействие происходит с посредством REST API.

Для редактирования, добавления, удаления рейсов используется REST API для сертифицированных авиаперевозчиков.

2.8.3 Требования к используемым технологиям

На стороне клиента используется фреймворк vue.js на языке программирования JavaScript.

На стороне сервера используется фреймворк Laravel на языке программирования php.

В роли БД будет использоваться MySQL в силу популярности и широты использования.

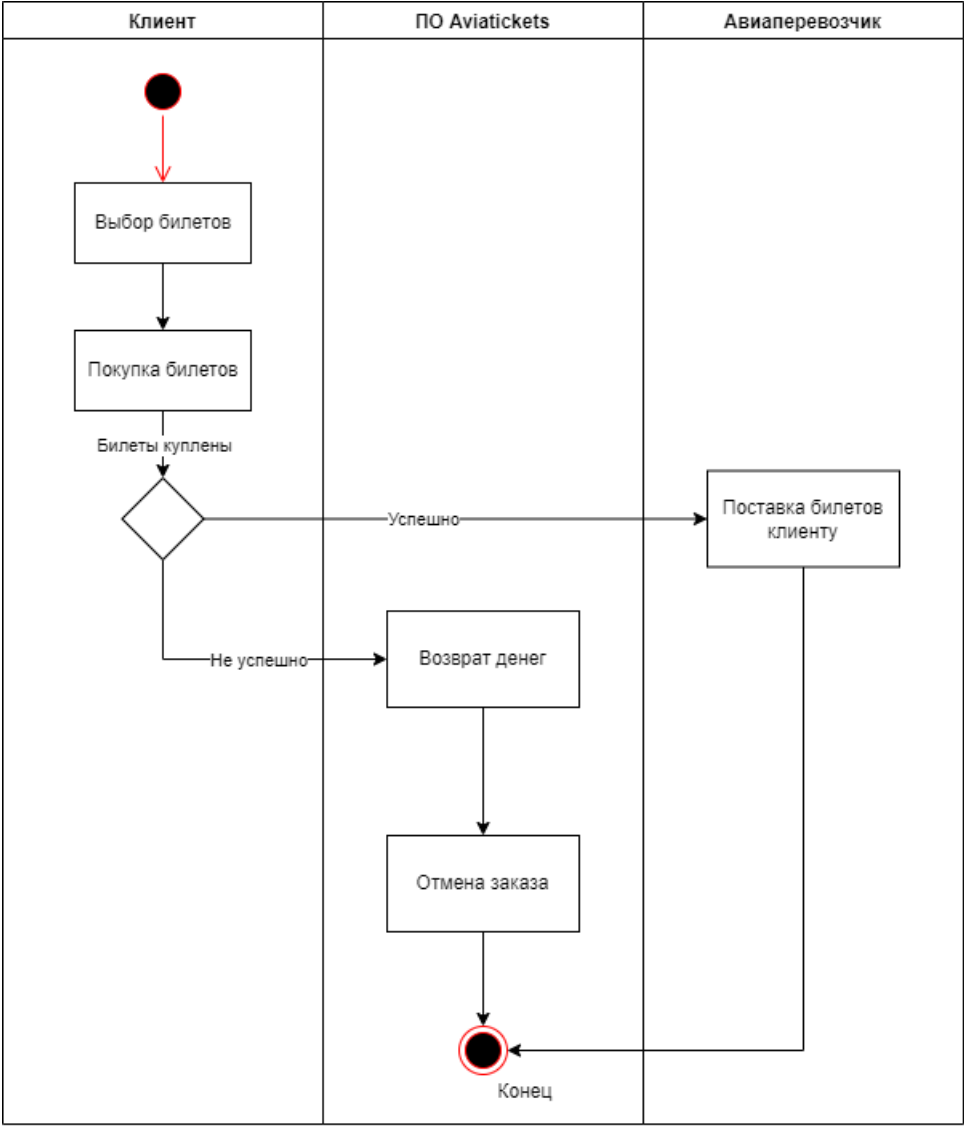
2.8.4 Дизайн веб-приложения

Текст должен быть хорошо виден на экране. Подсистема отображения пользовательского интерфейса и обработка действий пользователя должна обеспечивать удобный для конечного пользователя интерфейс, отвечающий следующим требованиям.

В части внешнего оформления:

- Интерфейсы должны быть выполнены в одной цветовой гамме и в едином стиле;
- Текст должен быть хорошо виден на экране;
- Интерфейсы должны быть на русском языке;
- При возникновении ошибок в работе подсистемы на экран монитора должно выводиться сообщение с наименованием ошибки.

2.9 Диаграмма активностей



2.10 Диаграмма последовательностей

