

Оглавление

Перечень используемых сокращений и аббревиатур	3
Глоссарий	3
1 Аналитический раздел	5
1.1 Анализ предметной области	5
1.2 Существующие решения.....	5
1.3 Процессы РСОИ	6
1.4 Участники РСОИ.....	10
1.5 Требования	11
1.5.1 Требования к РСОИ	11
1.5.2 Требования к диспетчерской системе	11
Функциональные требования с точки зрения пользователя	11
Входные параметры системы.....	12
Выходные параметры системы	13
1.5.3 Требования к АИС таксопарков.....	14
Требования к функциональным характеристикам	14
Функциональные требования с точки зрения пользователя	14
Входные параметры системы.....	15
Выходные параметры системы	15
1.5.4 Требования к АИС такси	16
Требования к функциональным характеристикам	16
Функциональные требования с точки зрения пользователя	16
Выходные параметры системы	16
1.5.5 Требования к протоколу взаимодействия между системами.....	17
1.6 Взаимодействия участников РСОИ	17
1.7 Сценарии функционирования	21
1.7.1 Спецификации прецедентов использования.....	22
2 Конструкторский раздел	26
2.1 Сущности заявок в системе	26
2.2 Взаимодействие подсистем	27
2.3 Протокол взаимодействия	29
2.4 Автоматы состояний заявки	32
2.5 Очереди в системе	35
3 Технологический раздел	36
3.1 Диаграмма БД	36
3.1.1 Диспетчерская АИС	36
3.1.2 АИС Таксопарка.....	37

3.1.3 АИС Такси	38
3.2 Язык программирования и среда разработки	38
3.3 Протоколы взаимодействия.....	39
3.3.1 Протокол синхронного взаимодействия	39
3.3.2 Протокол асинхронного взаимодействия	40
3.4 Система сборки.....	40
3.5 Сервер приложений.....	41
3.6 База данных	42
3.7 Развертывание системы	42
3.8 Кэширование.....	43
3.9 Интерфейс	44
3.10 Используемые библиотеки	48
3.11 Тестирование системы	49
3.11.1 Модульное тестирование.....	49
3.11.2 Системное тестирование	51

Перечень используемых сокращений и аббревиатур

БД – база данных

ПО – программное обеспечение

СПО – специальное программное обеспечение

РСОИ – распределённая система обработки информации

СУБД – система управления базами данных

АИС – автоматизированная информационная система

Глоссарий

РСОИ – система независимых (с точки зрения администрирования) взаимодействующих автоматизированных систем;

Программное обеспечение – совокупность программ системы обработки информации и документации, необходимой для эксплуатации этих программ;

Заявка – единица обслуживания внутри РСОИ.

Web-интерфейс – интерфейс пользователя, предоставляемый системой через web-браузер.

Узел – АИС, входящая в РСОИ.

Введение

Целью данной работы является создание распределенной системы обработки информации, позволяющей пользователям осуществлять заказ автомобилей такси посредством WEB - интерфейса. Поиск свободных таксистов осуществляется по нескольким таксопаркам с помощью фильтра поиска, задаваемого пользователем.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Анализ предметной области и происходящих в ней процессов, принятие решения об участниках РСОИ и их функциях;
2. Описание взаимодействия разрабатываемых систем;
3. Выделение основных составляющих в каждой системе и проектирования их взаимодействия;
4. Программная реализация спроектированных систем, модульное тестирование;
5. Системное тестирование всей распределённой системы в целом.

1 Аналитический раздел

В данном разделе анализируется предметная область, и выделяются требования к разрабатываемой РСОИ.

1.1 Анализ предметной области

Система заказа такси предназначена для подбора клиентам (пользователям системы) ближайших к ним свободных таксистов. Наибольшую сложность процесса заказа такси для клиента представляет поиск такси среди большого числа таксопарков. Клиенту необходимо связаться с несколькими таксопарками, чтобы сделать удовлетворяющий его выбор такси. Таксопарки, в свою очередь, заинтересованы в предоставлении своих услуг как можно большему числу клиентов. Чем ближе в данный момент времени находится свободный таксист к клиенту, сделавшему заказ, тем выгоднее как для клиента – такси приедет быстрее, так и для таксопарка – уменьшаются затраты на топливо для автомобилей. Каждая из организаций создает собственный узел проектируемой РСОИ – АИС, или подключается к существующему узлу. АИС разных организаций объединяются каналами связи и представляют собой системы, действующие независимо друг от друга. Проектируемая РСОИ позволит автоматизировать обработку заявок на подбор автомобилей, тем самым уменьшив человеческие трудозатраты.

1.2 Существующие решения

Крупнейшим существующим аналогом является сервис «Яндекс.Такси». Сервис позволяет вызывать официальное такси и наблюдать за его движением на карте. Вызов такси осуществляется через веб-интерфейс.

1.3 Процессы РСОИ

На рисунках 1-4 представлены основные процессы, происходящие в системе заказа такси.

На рисунке 1 представлена диаграмма для процесса заказа такси.

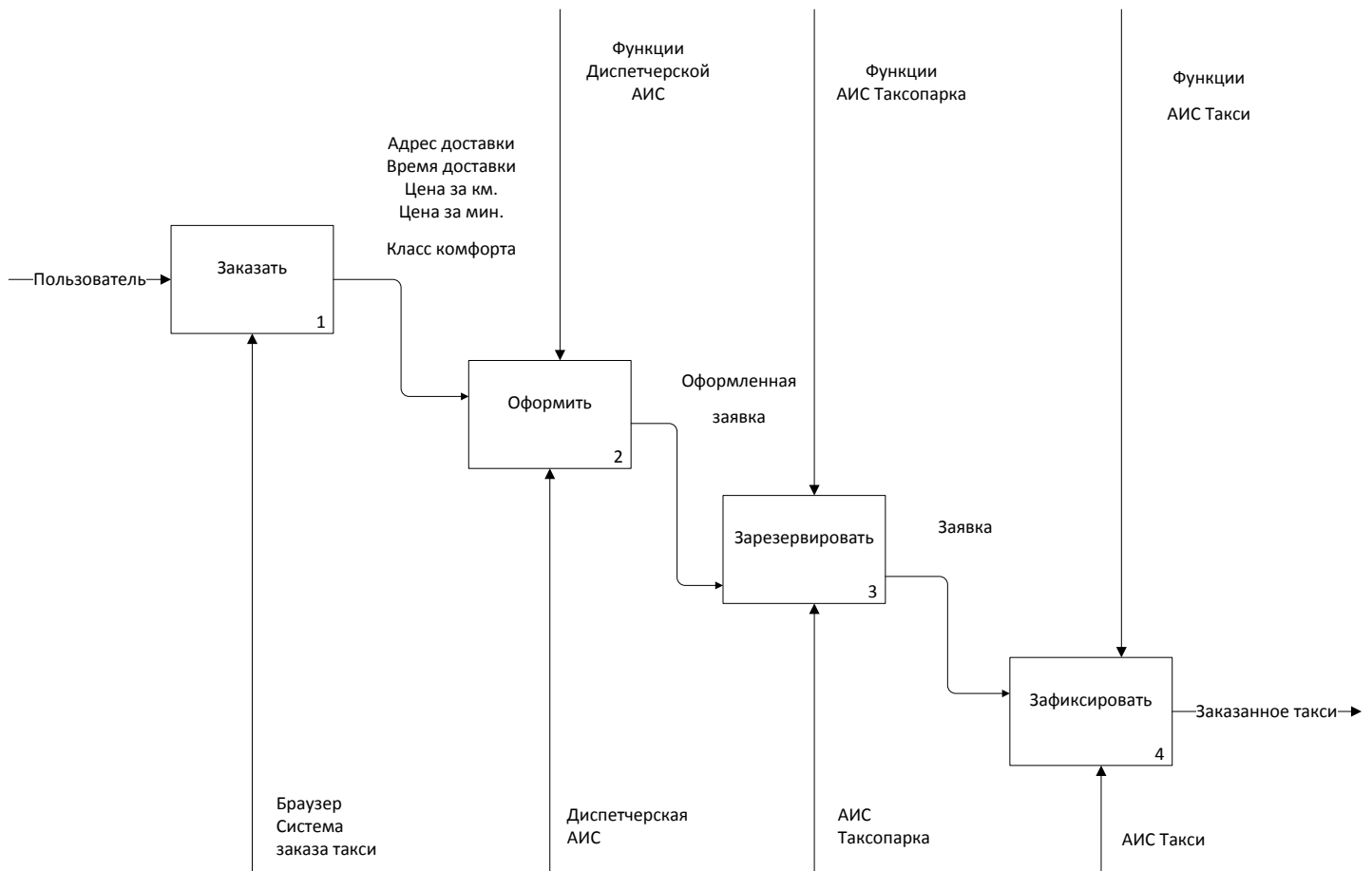


Рисунок 1. Диаграмма для процесса заказа такси.

На рисунке 2 представлена диаграмма для процесса получения текущего статуса заказа.

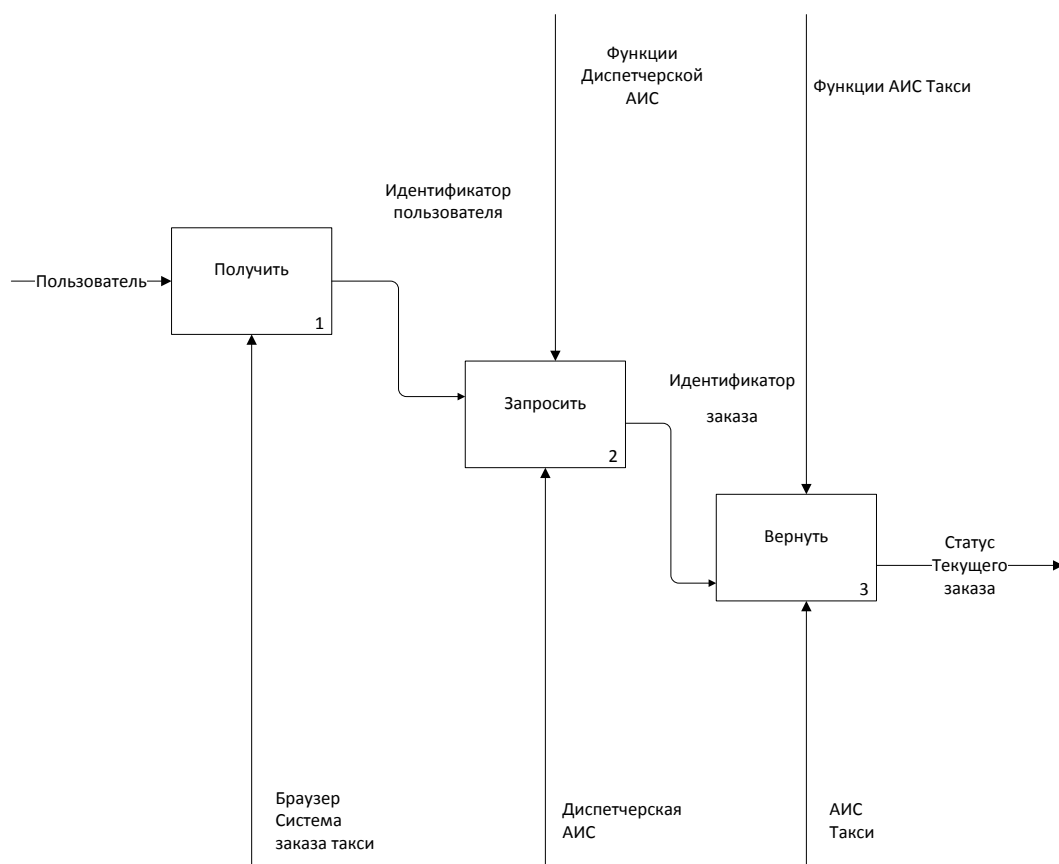


Рисунок 2. Диаграмма для процесса получения текущего статуса заказа.

На рисунке 3 представлена диаграмма для процесса получения истории предыдущих заказов пользователя:

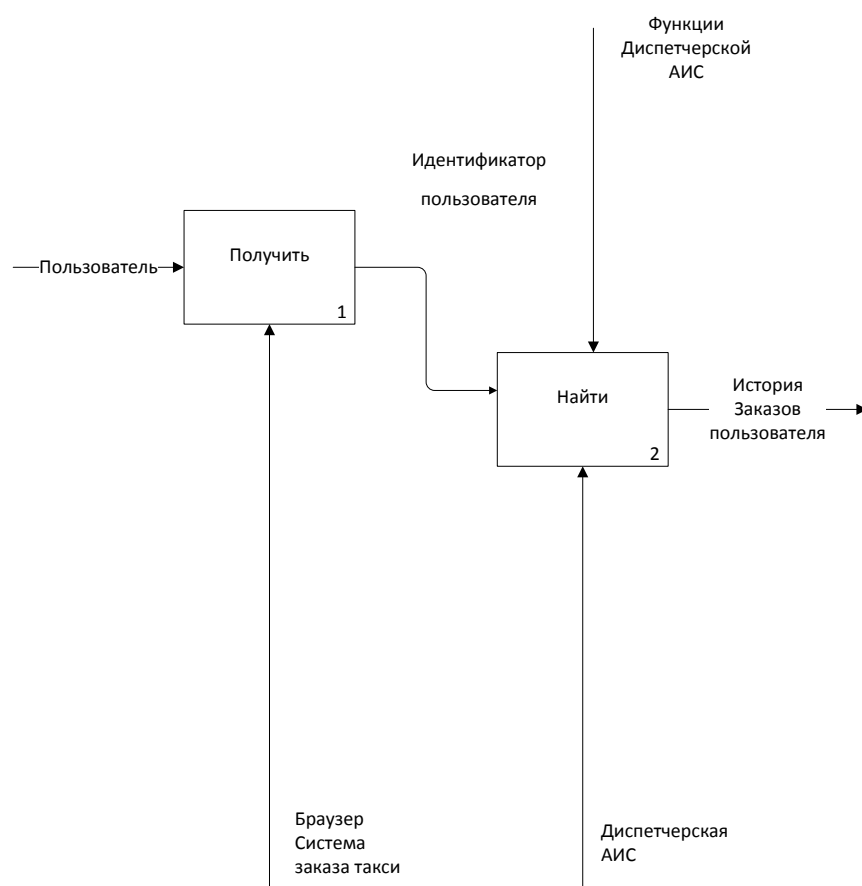


Рисунок 3. Диаграмма для процесса получения истории заказов пользователя.

На рисунке 4 представлена диаграмма для процесса отмены заявки:

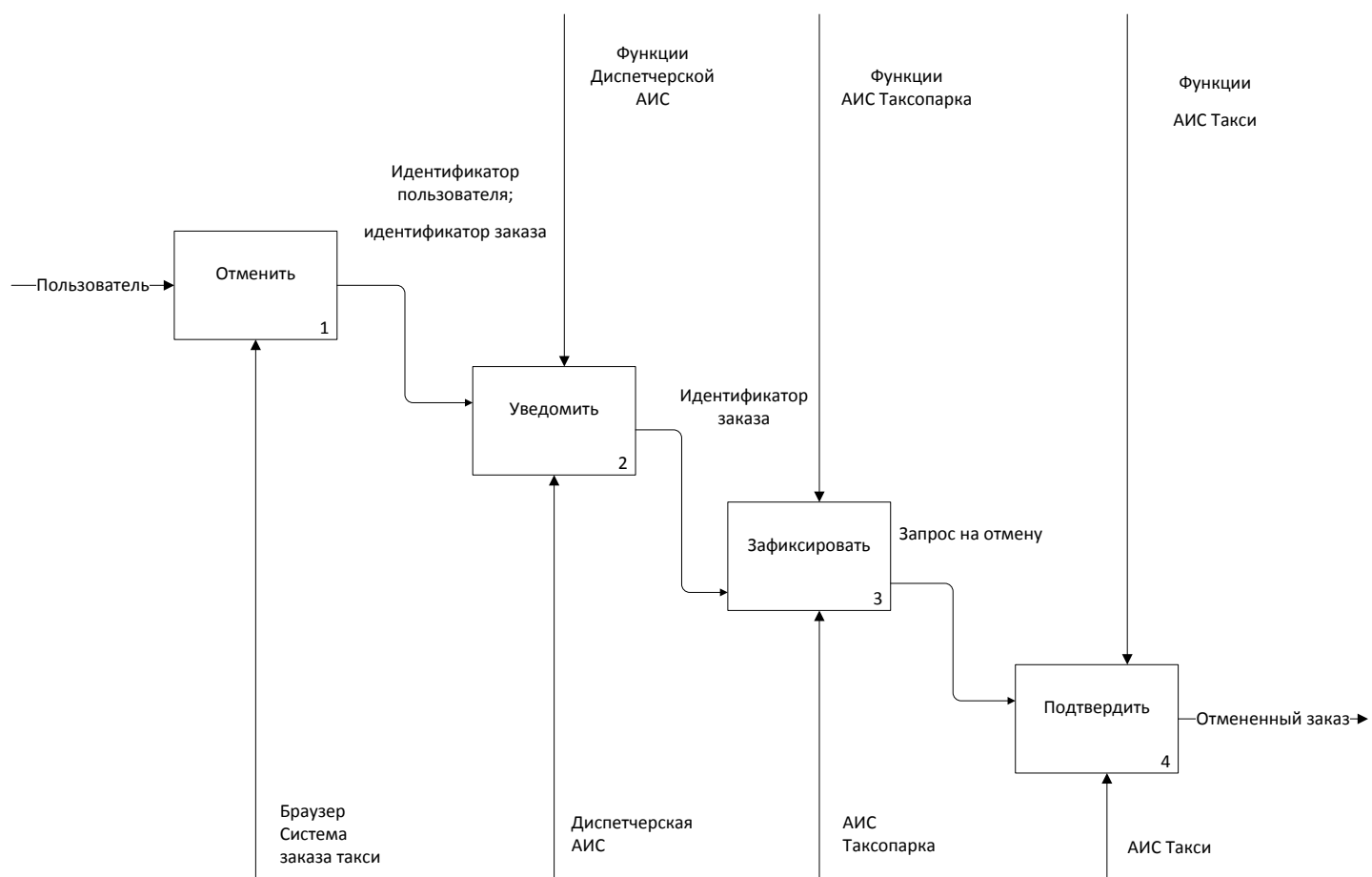


Рисунок 4. Диаграмма для процесса отмены заказа.

Общий вид модели предметной области представлен на Рисунке 5.

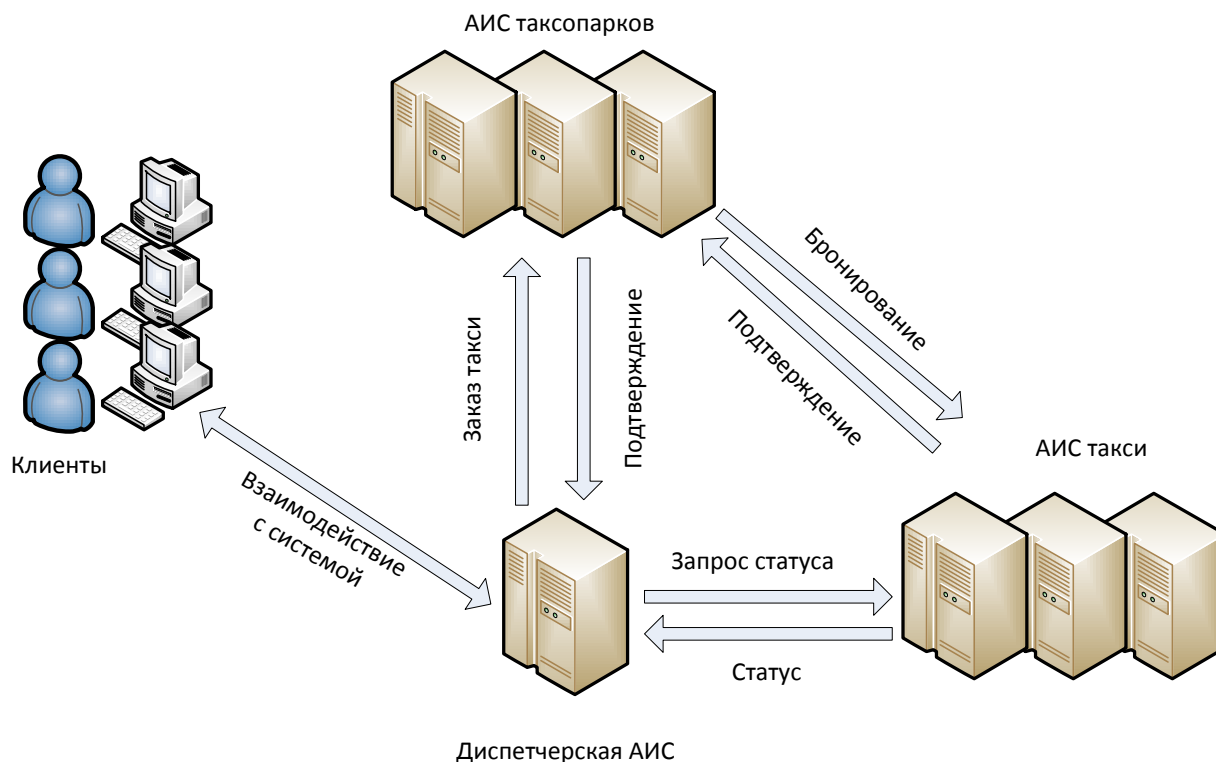


Рисунок 5. Модель предметной области

1.4 Участники РСОИ

Система заказа такси должна состоять из следующих узлов:

1. Диспетчерская АИС;
2. АИС в таксопарках;
3. АИС, устанавливающиеся на автомобили таксистов;

Основные задачи диспетчерской АИС – предоставление клиенту возможности заказа такси, удовлетворяющих его условиям заказа. Диспетчерская АИС обеспечивает связь с различными таксопарками, а также является связующим звеном между пользователем и таксистом, на которого была назначена заявка пользователя, для получения текущего статуса заказа.

Основные задачи АИС таксопарков – поиск свободных таксистов, зарегистрированных в данном таксопарке, готовых принять заказ. АИС

таксопарка назначает заявку на таксиста, при получении заявки от пользователя через диспетчерскую систему.

Основные задачи АИС такси – фиксирование заявки, назначенной АИС таксопарка, а также предоставление текущего статуса обработки заказа и координат диспетчерской АИС и АИС-ам таксопарков.

1.5 Требования

На основе проведенного анализа, могут быть сформулированы требования, предъявляемые как отдельным узлам, так и системе в целом.

1.5.1 Требования к РСОИ

Система должна иметь возможность добавления новых узлов.

Каждый узел системы должен иметь возможность конфигурирования для обеспечения взаимодействия с участниками РСОИ.

Узлы РСОИ должны проектироваться исходя из предположения об отсутствии абсолютно надежной связи.

Система должна предоставлять пользователю функции по формированию заявки на вызов такси.

Система должна предоставлять пользователю функции по просмотру истории предыдущих заказов.

Система должна предоставлять пользователю функции по просмотру статуса текущего заказа.

1.5.2 Требования к диспетчерской системе

Функциональные требования с точки зрения пользователя

Пользователями системы являются физические и юридические лица, желающие заказать такси.

Система должна предоставлять:

- web-интерфейс для взаимодействия;
- возможность регистрации и авторизации пользователей;
- авторизованному пользователю возможность перехода на страницу с историей предыдущих заказов;
- пользователю возможность выбора фильтра поиска такси по классу комфорта. Возможные значения: «Эконом», «Комфорт», «Бизнес»;
- пользователю возможность выбора фильтра поиска такси по цене за километр;
- пользователю возможность выбора фильтра поиска такси по цене за минуту;
- пользователю возможность осуществления заказа такси посредством web-интерфейса;
- авторизованному пользователю возможность перехода на страницу с отображением текущего статуса заказа;
- незарегистрированному пользователю возможность ввода кода доступа для получения текущего статуса заказа;

Система должна возвращать незарегистрированному пользователю код доступа для возможности получения текущего статуса его заказа.

Входные параметры системы

Система должна предоставлять возможность использования фильтра заказа такси по следующим полям, представленным в таблице 1:

Таблица 1. Параметры поиска Диспетчерской АИС.

Имя параметра	Тип параметра	Комментарии
---------------	---------------	-------------

Класс комфорта автомобиля	Список	Допустимые значения списка: «Эконом», «Комфорт», «Бизнес»
Верхняя граница цены за километр	Действительное число	
Верхняя граница цены за минуту	Действительное число	
Адрес доставки такси	Строка	
Время доставки такси	Дата	Точность до минут

Система должна предоставлять возможность регистрации нового пользователя с использованием следующих параметров:

Таблица 2. Параметры регистрации в Диспетчерской АИС.

Имя параметра	Тип параметра	Комментарии
Логин пользователя	Строковый	Обязательное поле
Пароль пользователя	Строковый	Обязательное поле
Имя пользователя	Строковый	Обязательное поле
Фамилия пользователя	Строковый	Необязательное поле
Отчество пользователя	Строковый	Необязательное поле

Система должна предоставлять возможность авторизации пользователя с использованием следующих параметров:

Таблица 3. Параметры авторизации в Диспетчерской АИС.

Имя параметра	Тип параметра	Комментарии
Логин пользователя	Строковый	Обязательное поле
Пароль пользователя	Строковый	Необязательное поле

Выходные параметры системы

Система должна выводить информацию в web-интерфейсе.

Система должна предоставлять возможность просмотра следующих параметров:

Таблица 4. Выходные параметры Диспетчерской АИС.

Имя параметра	Тип параметра	Комментарии
Текущий статус заявки	Строковый	
Текущий адрес автомобиля	Строковый	Данный параметр присутствует, если таксист принял заказ и выехал к клиенту
Код доступа для возможности просмотра текущего статуса заявки	Строковый	Предоставляется незарегистрированным пользователям
История предыдущих заказов	Список строк	Возможность просмотра истории предыдущих заказов предоставляется зарегистрированным пользователям

1.5.3 Требования к АИС таксопарков

Требования к функциональным характеристикам

Система должна предоставлять информацию о таксистах, удовлетворяющих условиям заявки клиента.

Система должна предоставлять возможность назначения таксистам нового задания.

Время отклика системы не должно превышать 5 секунд.

Функциональные требования с точки зрения пользователя

Пользователем системы является диспетчерская АИС.

Система должна предоставлять пользователю возможность формирования заявки на заказ такси.

Система должна предоставлять пользователю подтверждения заявки на заказ такси.

Система должна предоставлять пользователю возможность отмены заявки на заказ такси.

Система должна предоставлять пользователю возможность получения информации о таксисте, принявшем заказ.

Входные параметры системы

Список параметров, которые система должна принимать при поиске свободных таксистов:

Таблица 5. Входные параметры АИС Таксопарка.

Имя параметра	Тип параметра	Комментарии
Класс комфорта автомобиля	Строка	
Верхняя граница цены за километр	Действительное число	
Верхняя граница цены за минуту	Действительное число	
Адрес доставки такси	Строка	
Время доставки такси	Дата	Точность до минут
ФИО клиента	Строка	

Выходные параметры системы

Список параметров, которые система должна возвращать при постановке задачи таксисту.

Таблица 6. Выходные параметры АИС Таксопарка.

Имя параметра	Тип параметра	Комментарии
Марка автомобиля	Строка	
Идентификатор АИС такси	Строка	

1.5.4 Требования к АИС такси

Требования к функциональным характеристикам

Система должна предоставлять текущие координаты автомобиля.

Система должна предоставлять текущий статус такси.

Функциональные требования с точки зрения пользователя

Пользователем системы является диспетчерская АИС и АИС таксопарка.

Система должна обеспечивать пользователю возможность получения текущих координат такси.

Система должна обеспечивать пользователю возможность получения текущего статуса такси.

Выходные параметры системы

Таблица 7. Выходные параметры АИС Такси.

Имя параметра	Тип параметра	Комментарии
Текущий статус такси	Строковый	Может содержать следующие значения: «Выполняется заказ», «Свободен»
Текущие координаты	Строковый	Строка с координатами

автомобиля		автомобиля
------------	--	------------

1.5.5 Требования к протоколу взаимодействия между системами

Необходимо разработать протокол взаимодействия между системами, основанный на открытых стандартах. Под открытыми стандартами понимаются кроссплатформенные решения, не зависящие от какого-либо конкретного вендора.

Системы должны осуществлять передачу данных поверх протокола ТСР/ІР.

Передаваемое сообщение должно включать в себя следующие поля:

- уникальный идентификатор;
- тип;
- поля, зависящие от типа сообщения.

1.6 Взаимодействия участников РСОИ

В распределенной системе заказа такси на основе функциональных требований можно выделить следующие взаимодействия участников:

1. Клиент посредством WEB-интерфейса заполняет регистрационную форму и отправляет участнику «Диспетчерская АИС» запрос на регистрацию. «Диспетчерская АИС» обрабатывает запрос клиента и возвращает статус регистрации.
2. Клиент посредством WEB-интерфейса, заполнив форму логина, отправляет запрос участнику «Диспетчерская АИС» на авторизацию в системе. «Диспетчерская АИС» обрабатывает запрос клиента и возвращает статус авторизации.

3. Клиент заполняет поля фильтра заказа такси и посредством WEB-интерфейса отправляет заявку на заказ такси участнику РСОИ "Диспетчерская АИС". «Диспетчерская АИС» возвращает клиенту статус оформления заказа.
4. Клиент посредством WEB-интерфейса запрашивает текущий статус заказа у участника «Диспетчерская АИС». «Диспетчерская АИС» обрабатывает заявку клиента и возвращает статус текущего заказа.
5. Клиент посредством WEB-интерфейса запрашивает историю предыдущих заказов у участника «Диспетчерская АИС». «Диспетчерская АИС» возвращает клиенту историю предыдущих заказов.
6. «Диспетчерская АИС» посылает заявку на заказ такси всем зарегистрированным в ней «АИС Таксопарка». «АИС Таксопарка» возвращают статусы обработки заявок от «Диспетчерской АИС», сообщая о наличии свободных таксистов, удовлетворяющих фильтрам поиска, или их отсутствии.
7. «АИС Таксопарка» опрашивает все зарегистрированные в данном таксопарке «АИС Такси», удовлетворяющие фильтрам заказа, на возможность принять заказ от клиента. «АИС Такси» возвращают «АИС Таксопарка» текущий статус, сообщая о готовности принять заказ.
8. «Диспетчерская АИС» посылает заявку на оформление заказа в «АИС Таксопарка» со свободными «АИС такси». «АИС Таксопарка» возвращает «Диспетчерской АИС» статус оформления заказа и данные о принявшем заказ участнике «АИС Такси» для возможности установления связи напрямую между «Диспетчерской АИС» и «АИС Такси». «Диспетчерская АИС» запускает кэширование текущего статуса заказа в локальной БД.

9. Клиент посредством WEB-интерфейса запрашивает текущий статус заказа у участника «Диспетчерская АИС». «Диспетчерская АИС» возвращает статус заказа из локальной БД.
10. Клиент посредством WEB-интерфейса запрашивает историю всех своих заказов у участника «Диспетчерская АИС». «Диспетчерская АИС» возвращает историю заказов клиента из локальной БД.
11. Клиент посредством WEB-интерфейса отменяет текущий заказ, посылая запрос на отмену участнику «Диспетчерская АИС». «Диспетчерская АИС» посылает запрос на отмену заказа в «АИС Таксопарка», в которой был выполнен заказ. «АИС Таксопарка» посылает запрос на отмену «АИС Такси», на которую зарегистрирован заказ. «АИС Такси» возвращает статус отмены заказа в «АИС Таксопарка». «АИС Таксопарка» возвращает статус отмены заказа в «Диспетчерскую АИС». «Диспетчерская АИС» изменяет статус заказа в локальной БД.

На рисунке 6 представлена последовательность действий при заказе такси.

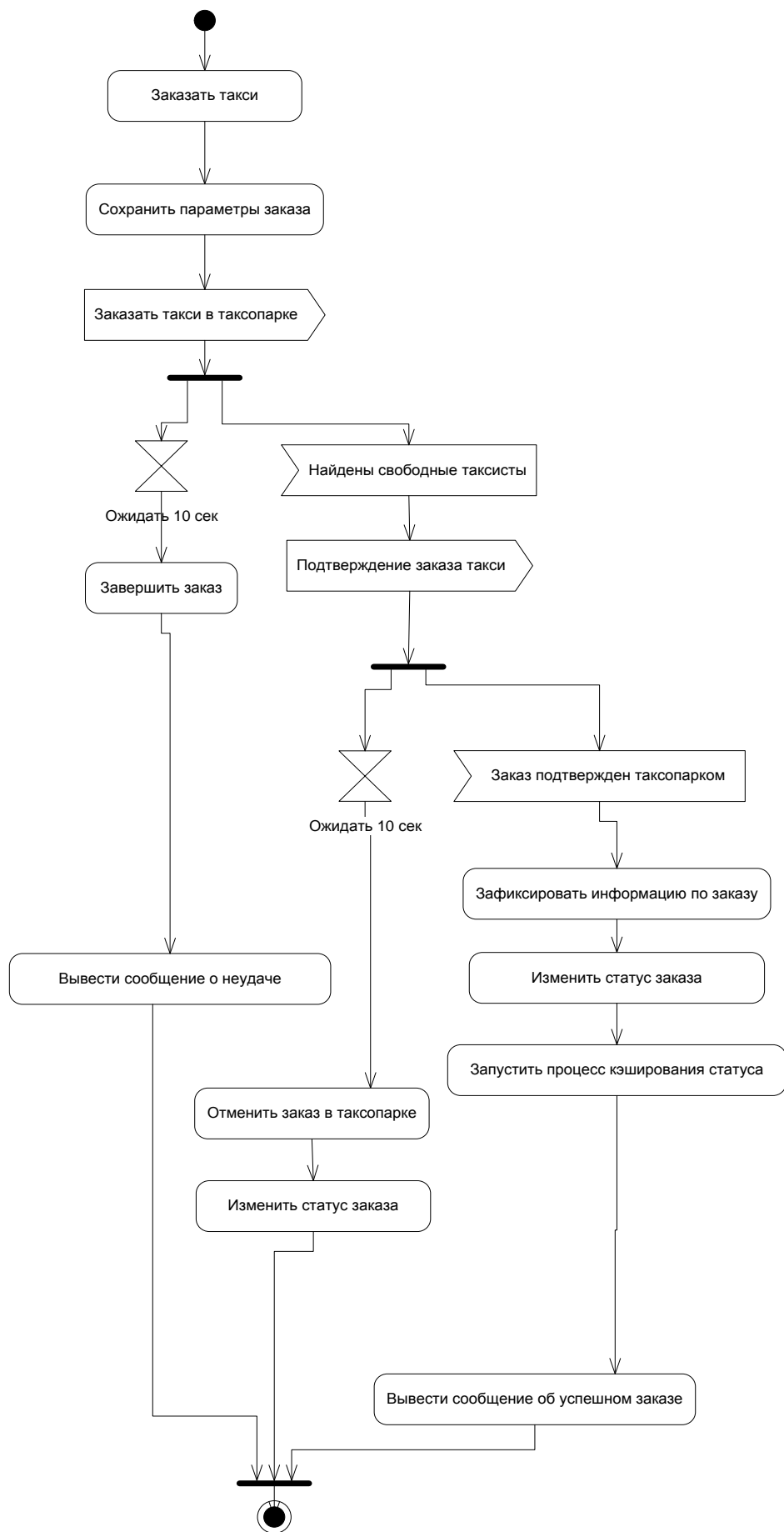


Рисунок 6. Последовательность действий при заказе такси.

1.7 Сценарии функционирования

Для выполнения любого сценария необходимо войти в систему через интерфейс пользователя. В системе допустимы 2 типа пользователей – гость и зарегистрированный пользователь. Сценарии функционирования для пользователей с категорией прав «Гость» и «Зарегистрированный пользователь» отражены на рисунке 7 и рисунке 8 соответственно.

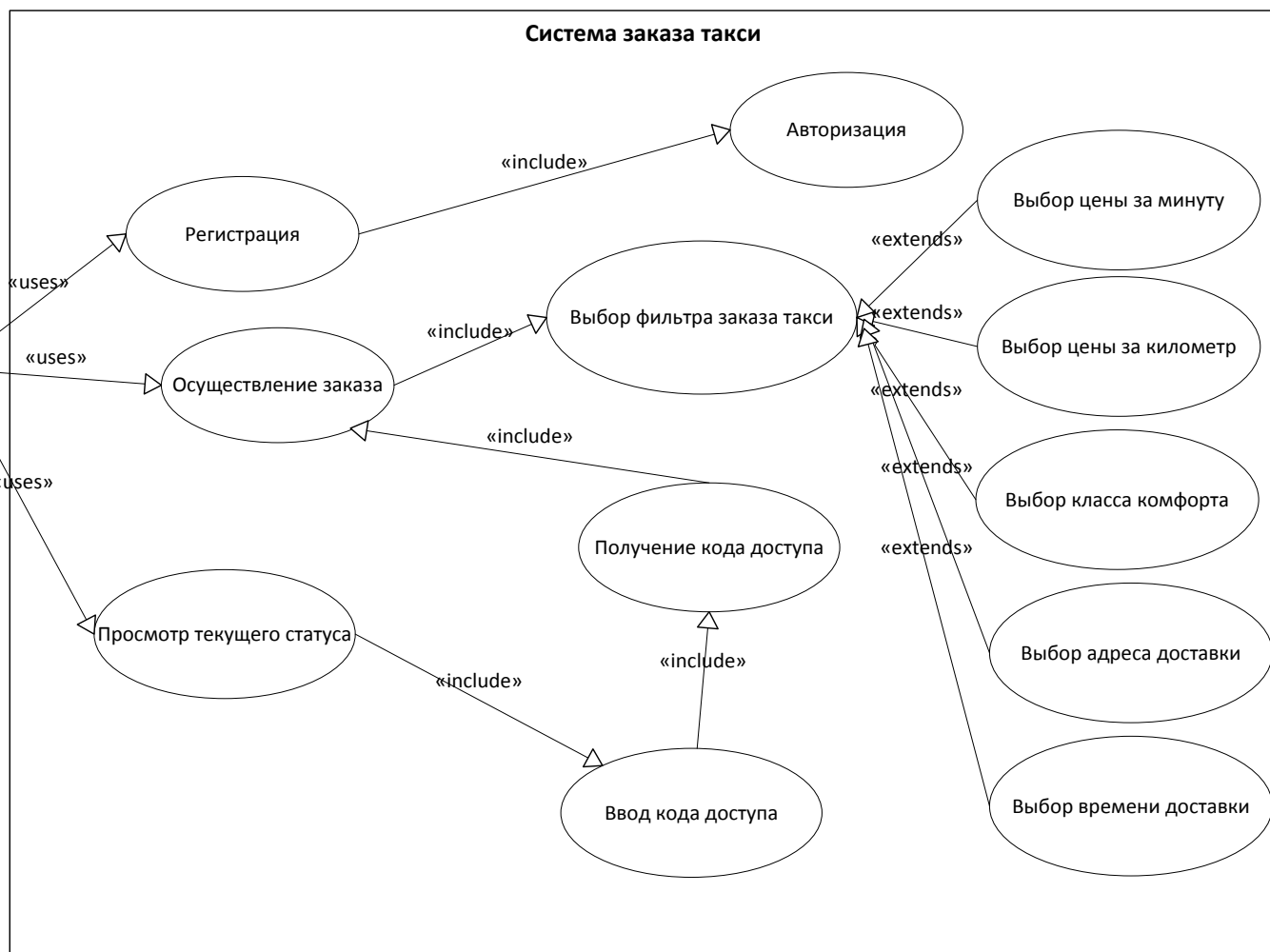


Рисунок 7. Диаграмма использования для пользователя с категорией прав «Гость».

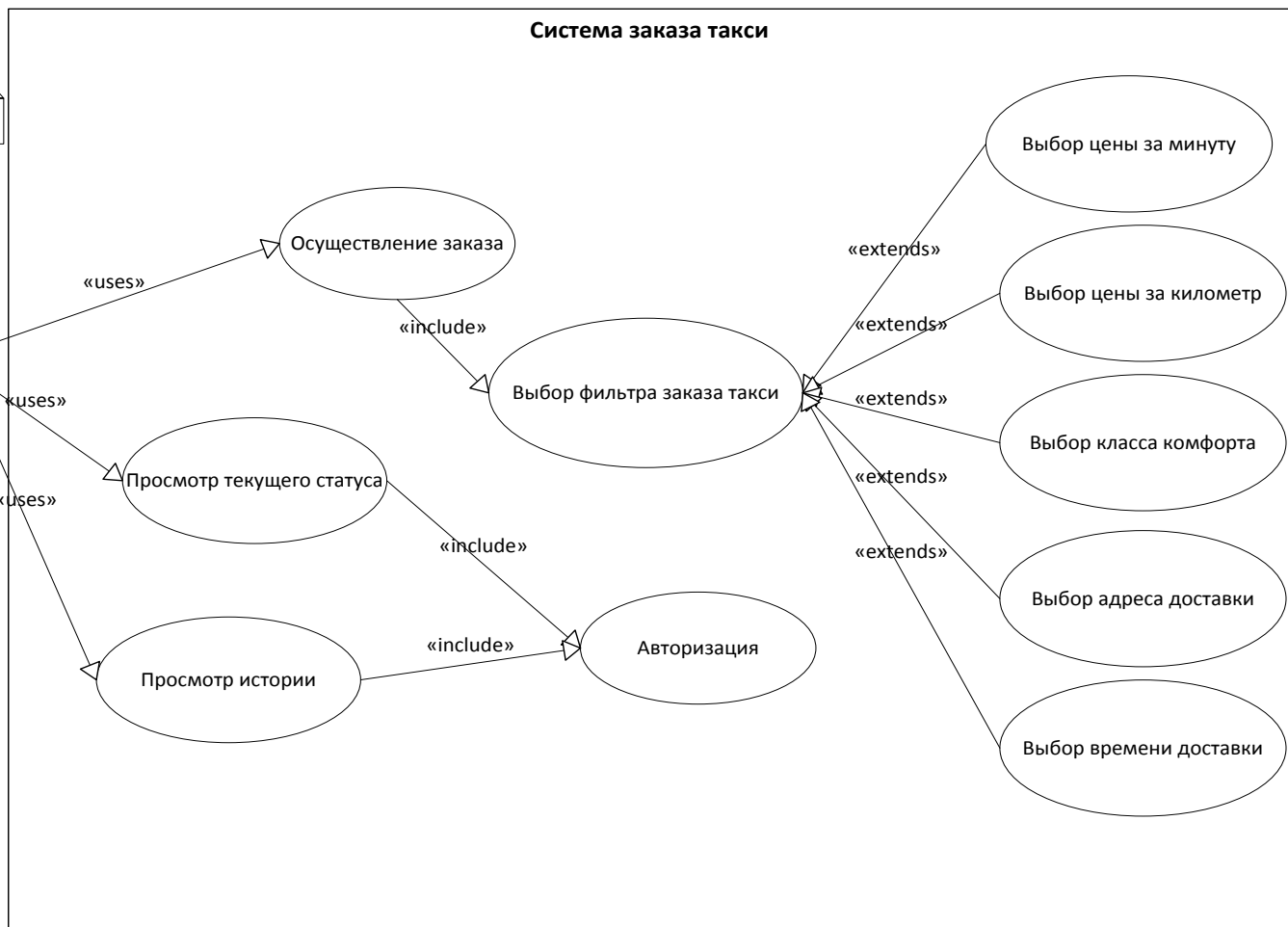


Рисунок 8. Диаграмма использования для пользователя с категорией прав «Зарегистрированный пользователь».

1.7.1 Спецификации прецедентов использования

Спецификация прецедента «Регистрация»

Прецедент использования позволяет действующему лицу зарегистрироваться в системе.

Сценарий:

1. Выполнить вход в web-интерфейс.
2. В web-интерфейсе перейти к регистрации по ссылке.
3. Заполнить поля «Логин пользователя», «Пароль», «Фамилия», «Имя», «Отчество».
4. Зарегистрироваться, нажав кнопку «Регистрация» или отказаться от регистрации, нажав кнопку «Отмена».

Спецификация прецедента «Авторизация»

Прецедент использования позволяет действующему лицу войти в систему.

Сценарий:

Основной сценарий, возможен, если пользователь является зарегистрированным пользователем

1. Выполнить вход в web-интерфейс.
2. В web-интерфейсе перейти ко входу по ссылке.
3. Заполнить поля «Логин» и «Пароль».
4. Подтвердить вход в систему, нажав кнопку «Войти» или отменить вход, нажав кнопку «Отмена».

Альтернативный сценарий: если пользователь не является зарегистрированным пользователем, то необходимо вначале выполнить регистрацию и попробовать войти в систему после.

Спецификация прецедента «Осуществление заказа»

Прецедент использования позволяет действующему лицу сформировать заявку на заказ такси.

Основной сценарий: возможен, если пользователь имеет категорию прав «Зарегистрированный пользователь»

1. Выполнить вход в web-интерфейс.
2. Перейти на страницу заказа такси.
3. Заполнить поля фильтра заказа такси: «Класс комфорта», «Цена за километр», «Цена за минуту», «Адрес доставки», «Время доставки»
4. Выполнить заказ такси, нажав кнопку «Заказать».

Альтернативный сценарий, если пользователь имеет категорию прав «Гость»

1. Выполнить вход в web-интерфейс.
2. Перейти на страницу заказа такси.
3. Заполнить поля фильтра заказа такси: «Класс комфорта», «Цена за километр», «Цена за минуту», «Адрес доставки», «Время доставки»
4. Выполнить заказ такси, нажав кнопку «Заказать».
5. Получить код доступа для возможности просмотра статуса заказа
6. Сохранить код доступа

Спецификация прецедента «Просмотр текущего статуса»

Прецедент использования позволяет пользователю посмотреть текущий статус сделанного заказа.

Основной сценарий: возможен, когда пользователь имеет категорию прав «Зарегистрированный пользователь»

1. Выполнить вход в web-интерфейс.
2. Если вход в систему не выполнен, выполнить авторизацию.
3. Перейти на страницу просмотра текущего статуса заказа.

Альтернативный сценарий, если пользователь имеет категорию прав «Гость»

1. Выполнить вход в web-интерфейс.
2. Перейти на страницу просмотра текущего статуса заказа.
3. Ввести сохраненный код доступа в поле «Код для просмотра статуса»
4. Нажать кнопку «Посмотреть статус»

Спецификация прецедента "Просмотр истории"

Прецедент использования позволяет зарегистрированному пользователю посмотреть историю предыдущих заказов.

Основной сценарий: возможен, когда пользователь имеет категорию прав «Зарегистрированный пользователь».

Выполнить вход в web-интерфейс.

Если вход в систему не осуществлен, выполнить авторизацию.

Перейти на страницу просмотра истории предыдущих заказов.

2 Конструкторский раздел

2.1 Сущности заявок в системе

На рисунке 9 представлена ER диаграмма Диспетчерской АИС.

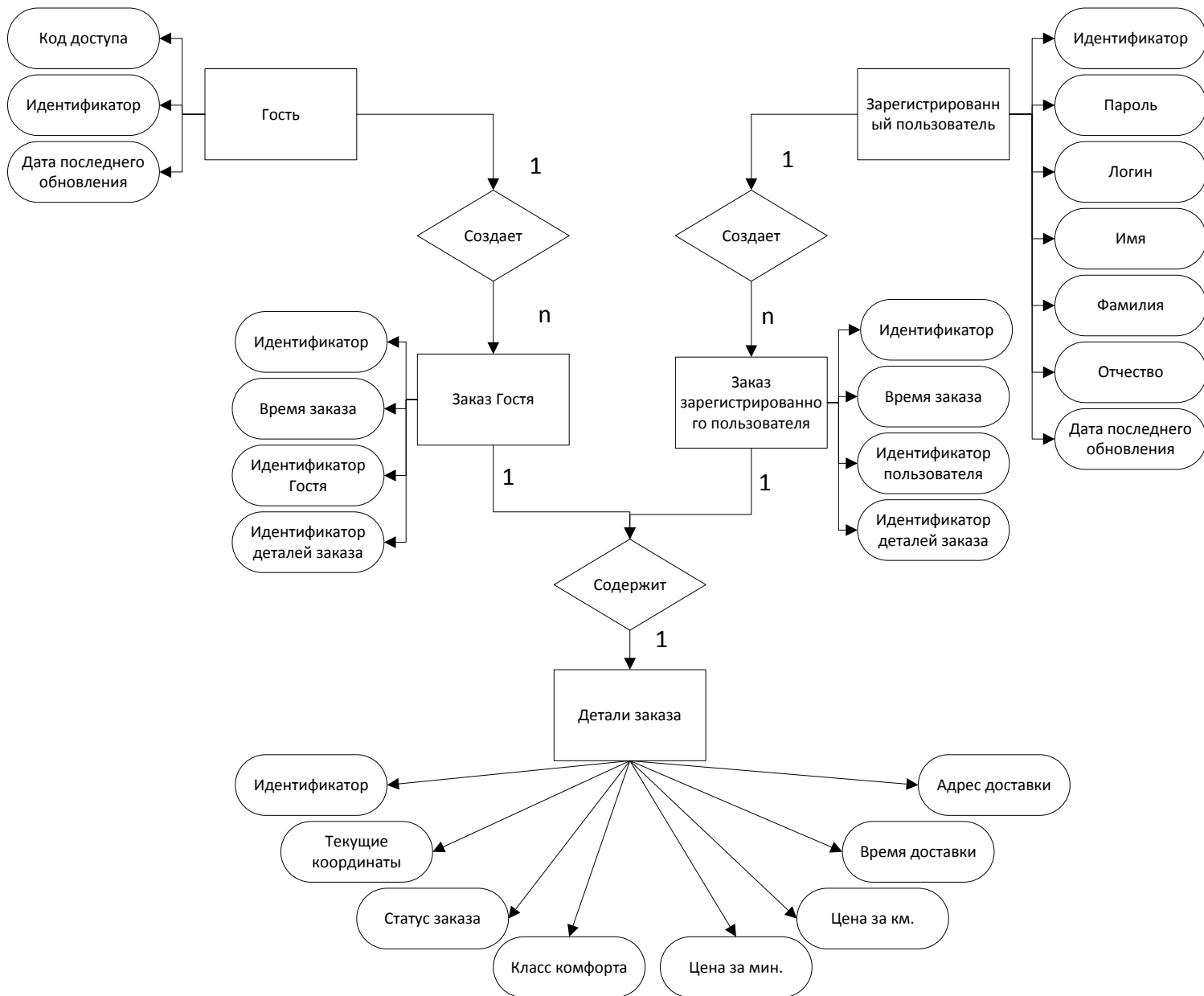


Рисунок 9. ER диаграмма Диспетчерской АИС.

2.2 Взаимодействие подсистем

На рисунке 10 представлен случай заказа автомобиля такси. Диспетчерская система получает от клиента запрос на заказ автомобиля такси с набором параметров заказа. Диспетчерская система посылает запрос на заказ всем зарегистрированным в ней подсистемам таксопарков. Подсистема таксопарка выполняет фильтрацию зарегистрированных в ней таксистов по параметрам, указанным пользователем. Если не найдено автомобилей такси, удовлетворяющих условиям поиска, подсистема таксопарка возвращает Диспетчерской системе сообщение о невозможности выполнения заказа. В противном случае, подсистема таксопарка посылает запрос на получение текущего статуса во все зарегистрированные в ней подсистемы такси. Подсистема такси возвращает текущий статус. Если имеются свободные таксисты, подсистема таксопарка сообщает диспетчерской системе о возможности совершить заказ. Диспетчерская система подтверждает совершение заказа. Подсистема таксопарка регистрирует новую задачу для первой ответившей подсистемы такси со статусом «свободен» и возвращает диспетчерской системе идентификатор подсистемы такси, принявшей заказ, для возможности дальнейшего получения статуса заказа через подсистему такси напрямую.

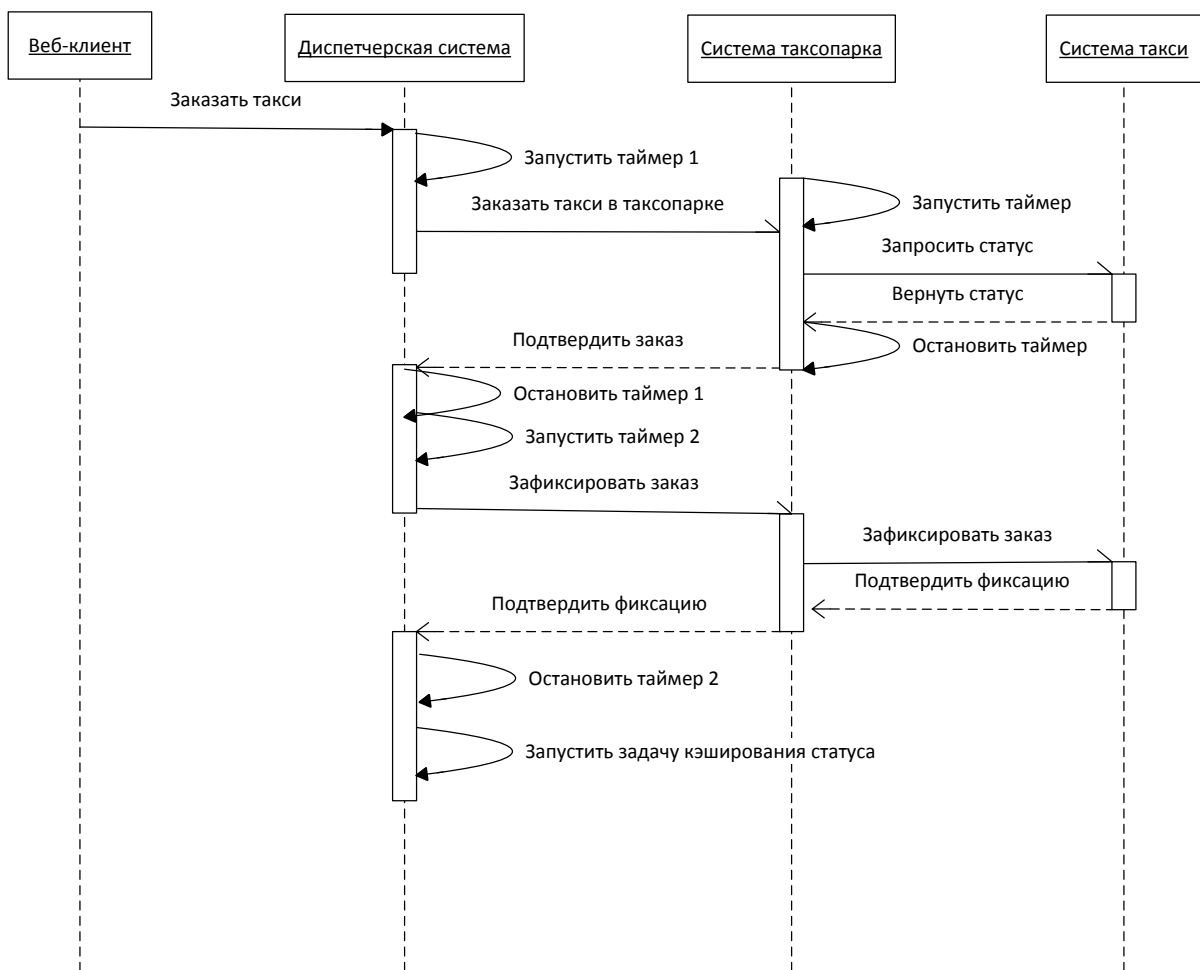


Рисунок 10. Заказ автомобиля такси.

2.3 Протокол взаимодействия

Сообщение на заказ такси:

При осуществлении заказа такси в Диспетчерскую систему асинхронно направляется сообщение, содержащее параметры заказа – адрес доставки, время доставки, цена за километр, цена за минуту, класс комфорта и идентификатор пользователя (в случае зарегистрированного в системе пользователя). Диспетчерская система отправляет заявки на заказ в очереди зарегистрированных в ней таксопарков. Диспетчерская АИС «слушает» очереди входящих сообщений от АИС Таксопарков и от АИС Такси.

Система таксопарка получает сообщение из очереди входящих от Диспетчерской системы сообщений и выполняет поиск таксистов, удовлетворяющих параметрам запроса в локальной БД. Система таксопарка направляет сообщения на получение статуса в очереди АИС Такси. Система Таксопарка «слушает» очередь входящих сообщений от Диспетчерской АИС и очередь входящих сообщений от АИС Такси.

АИС Такси получает сообщение из очереди входящих от Диспетчерской АИС сообщений и возвращает ответ Диспетчерской АИС.

АИС Такси получает сообщение из очереди входящих от АИС Таксопарка сообщений и возвращает ответ АИС Таксопарка.

Сообщение получения истории заказов:

Для получения истории заказов пользователя, в Диспетчерскую АИС отправляется синхронное сообщение с идентификатором пользователя. Диспетчерская АИС возвращает историю заказов пользователя из локальной БД.

Сообщение получения статуса текущего заказа:

Для получения статуса текущего заказа, в Диспетчерскую АИС отправляется синхронное сообщение с идентификатором пользователя (в случае зарегистрированного пользователя), либо с кодом доступа (если пользователь не зарегистрирован в системе). Диспетчерская АИС возвращает текущий статус заказа из локальной БД (При фиксации заказа в Диспетчерской АИС запускается задача кэширования текущего статуса заказа в локальной БД). Под статусом заказа понимается текущее расположение заказанного автомобиля такси, а также приблизительное время, оставшееся до прибытия автомобиля такси по месту заказа.

Сообщения, передаваемые между системами, их типы и параметры сведены в таблице 8:

Таблица 8. Передаваемые в системе заказа такси сообщения.

Сообщение	Отправитель	Получатель	Тип запроса	Параметры запроса	Ответ
Заказ такси	Пользователь	Диспетчерская АИС	Асинхронный	идентификатор пользователя; адрес доставки; время доставки; цена за мин.; цена за км.; класс комфорта.	Сообщение об удачном принятии заказа в обработку, или сообщение об ошибке.
Запрос на наличие подходящих таксистов	Диспетчерская АИС	АИС Таксопарка	Асинхронный	Цена за мин; Цена за км; Класс комфорта.	Сообщение о наличии или отсутствии подходящих таксистов.
Запрос статуса таксиста	АИС Таксопарка	АИС Такси	Асинхронный		Текущий статус таксиста
Подтверждение	Диспетчерская	АИС	Асинхронный	Идентификато	Сообщение

е заказа	АИС	Таксопарка		р заказа	об успешном подтверждении заказа и идентификатор такси или сообщение о неудаче
Формирование новой задачи	АИС Таксопарка	АИС Такси	Асинхронный	ФИО пользователя; Адрес доставки; Время доставки;	Сообщение об успешном принятии заказа в обработку или сообщение о неудаче
Запрос статуса	Пользователь	Диспетчерская АИС	Синхронный	Идентификатор пользователя	Статус текущего заказа пользователя
Запрос истории	Пользователь	Диспетчерская АИС	Синхронный	Идентификатор пользователя	История заказов данного пользователя
Запрос на отмену текущего заказа	Пользователь	Диспетчерская АИС	Асинхронный	Идентификатор пользователя; Идентификатор заказа.	Сообщение об успешном принятии в обработку запроса на отмену заказа или сообщение о неудаче

2.4 Автоматы состояний заявки

На рисунке 11 представлен автомат состояний заявки в Диспетчерской АИС. Заявка считается успешной, если она прошла через состояние «Успешное завершение».

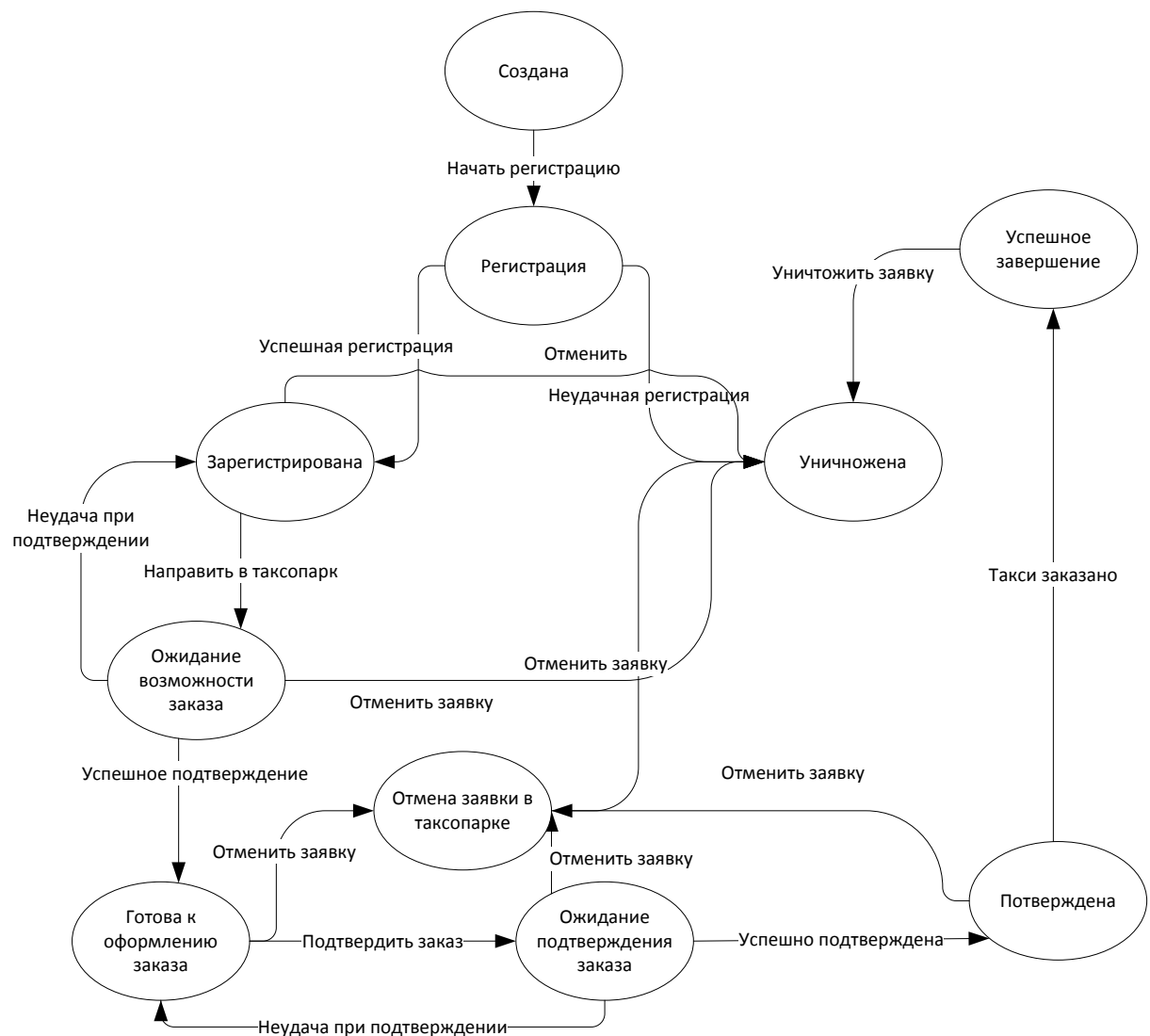


Рисунок 11. Автомат состояний заявки в Диспетчерской АИС.

На рисунке 12 представлен автомат состояний заявки в АИС Таксопарк. Заявка считается успешной, если она прошла через состояние «Успешное завершение».

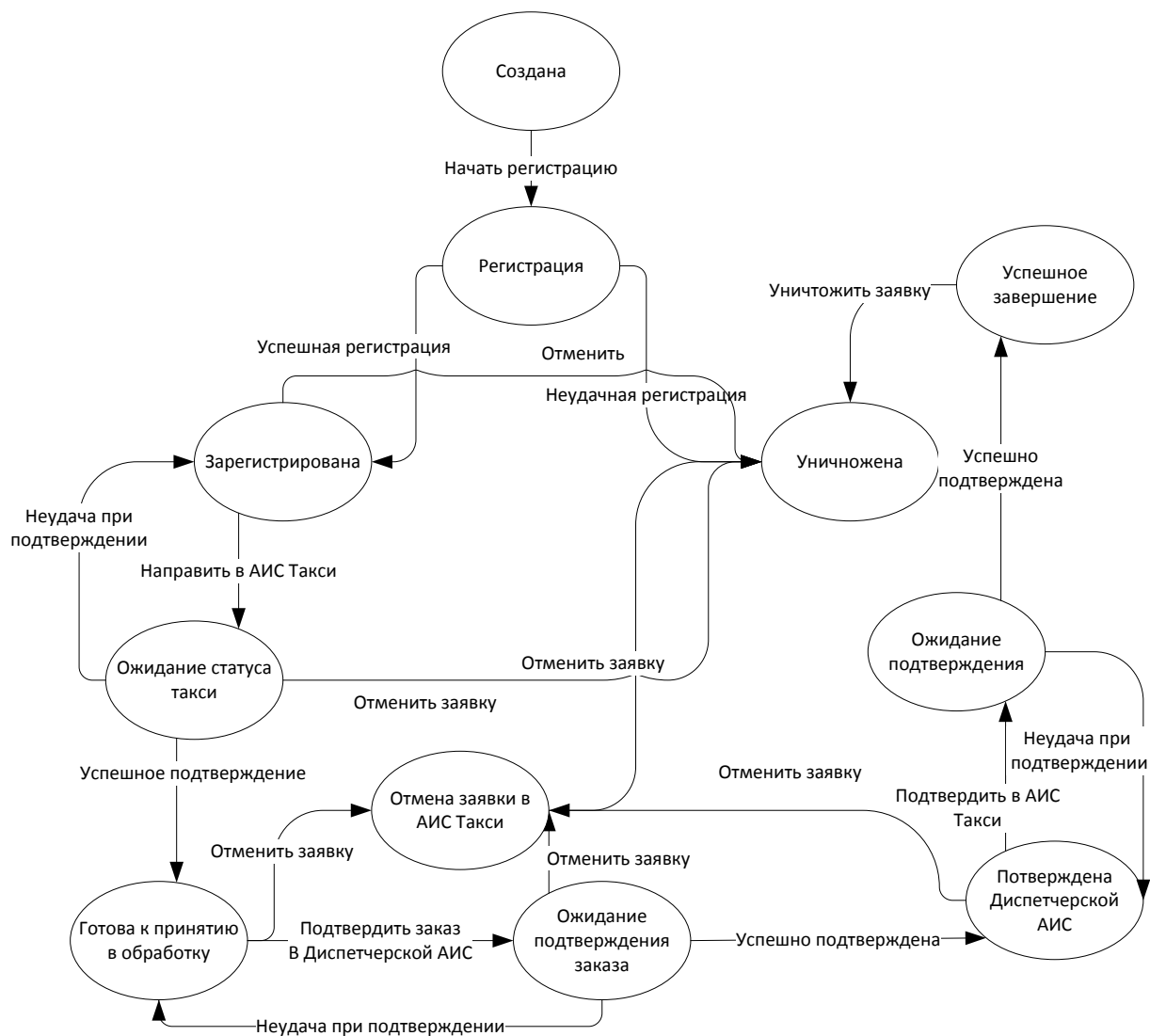


Рисунок 12. Автомат состояний заявки в АИС Таксопарка.

На рисунке 13 представлен автомат состояний заявки в АИС Такси. Заявка считается успешной, если она прошла через состояние «Успешное завершение».

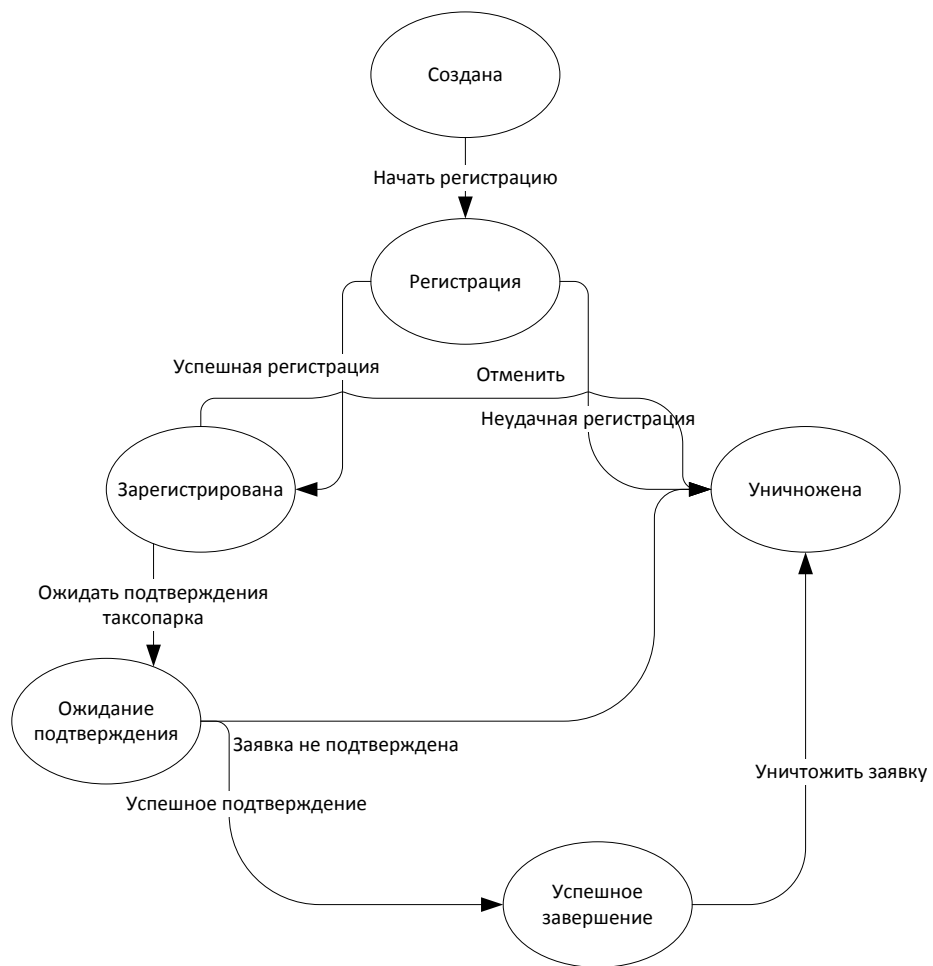


Рисунок 13. Автомат состояний заявки в АИС Такси.

2.5 Очереди в системе

При реализации курсового проекта на транспортном сервере для каждой из систем были зарегистрированы очереди сообщений. Были развернуты:

- **Диспетчерская система.**

«Слушает» очереди DISP.TAXI.IN для получения входящих сообщений от АИС Такси и DISP.TAXIPARK.IN для получения входящих сообщений от АИС Таксопарков.

- **2 Системы Таксопарков.**

«Слушают» очереди TAXIPARK1.DISP.IN и TAXIPARK2.DISP.IN для получения входящих сообщений от Диспетчерской АИС. Очереди TAXIPARK1.TAXI.IN, TAXIPARK2.TAXI.IN для получения входящих сообщений от АИС Такси.

- **6 систем Такси (по 3 для каждого таксопарка).**

«Слушают» очереди TAXI1.TAXIPARK1.IN,
TAXI2.TAXIPARK1.IN, TAXI3.TAXIPARK1.IN,
TAXI1.TAXIPARK2.IN, TAXI2.TAXIPARK2.IN,
TAXI3.TAXIPARK2.IN на получение входящих сообщений от АИС Таксопарков и от Диспетчерской АИС.

3 Технологический раздел

3.1 Диаграмма БД

3.1.1 Диспетчерская АИС

На рисунке 14 представлена схема БД для Диспетчерской АИС:

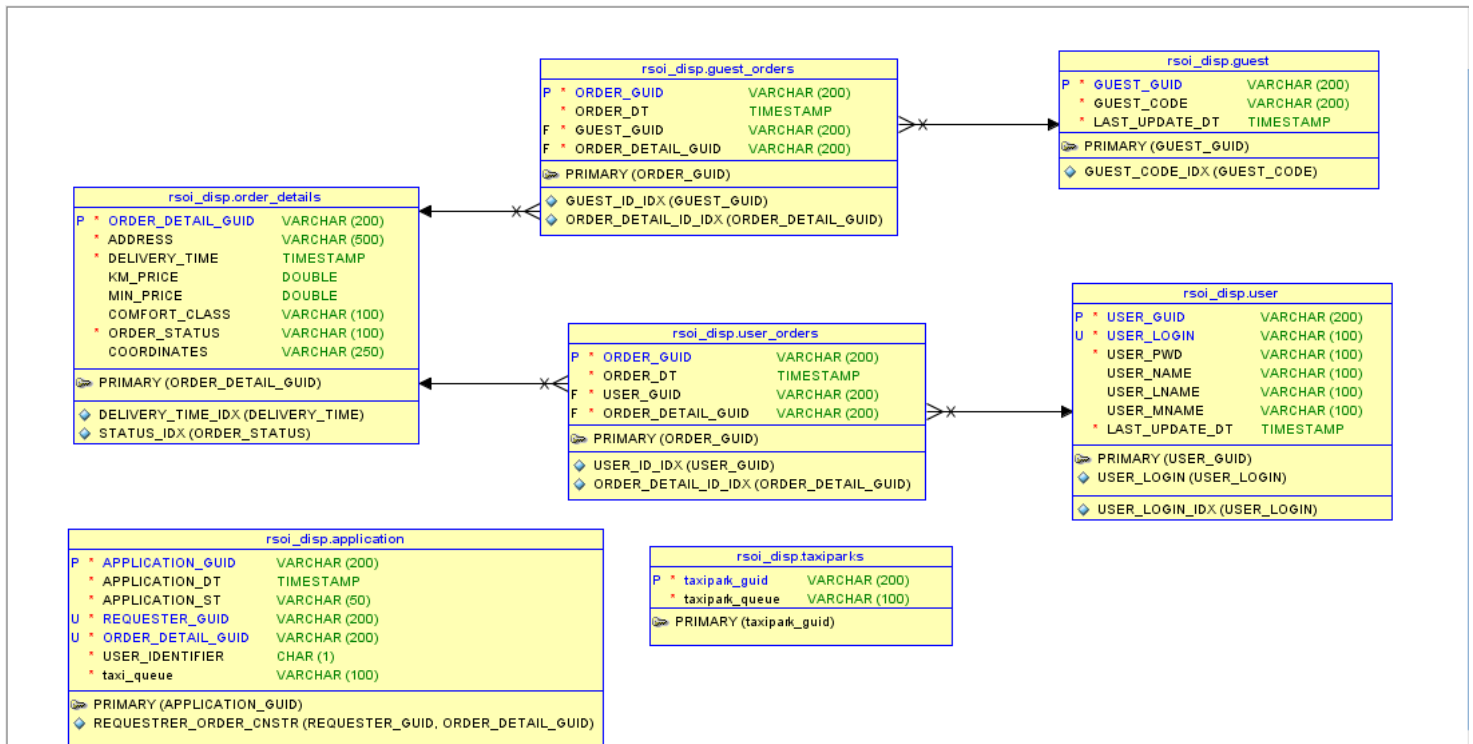


Рисунок 14. Схема БД Диспетчерской АИС

При поступлении заявки на заказ такси от клиента, создается новая заявка в таблице ORDER_DETAILS с параметрами фильтра, заполненными клиентом при заказе такси. Добавляется новый заказ в таблицу APPLICATION со ссылкой на таблицу ORDER_DETAILS.

При подтверждении заказа в таблице APPLICATION статус заказа меняется на “CONFIRMED” и по срабатыванию триггера заказ переносится в таблицу USER_ORDERS или GUEST_ORDERS в зависимости от того, зарегистрирован ли пользователь в системе.

Таблица USER содержит всех пользователей, зарегистрированных в системе, их идентификаторы и данные для авторизации.

Таблице GUEST содержит идентификаторы незарегистрированных в системе пользователей и их коды доступа для получения текущего статуса заказа.

При поступлении запрос от клиента на получение истории предыдущих заказов поиск истории осуществляется по таблице USER_ORDERS с помощью переданного в сообщении идентификатора пользователя.

Таблица TAXIPARKS содержит все зарегистрированные в Диспетчерской АИС Таксопарки – их идентификаторы и идентификаторы очередей, которые они «слушают» для возможности обмена сообщениями с ними.

3.1.2 АИС Таксопарка

На рисунке 15 представлена диаграмма БД для АИС Таксопарка:

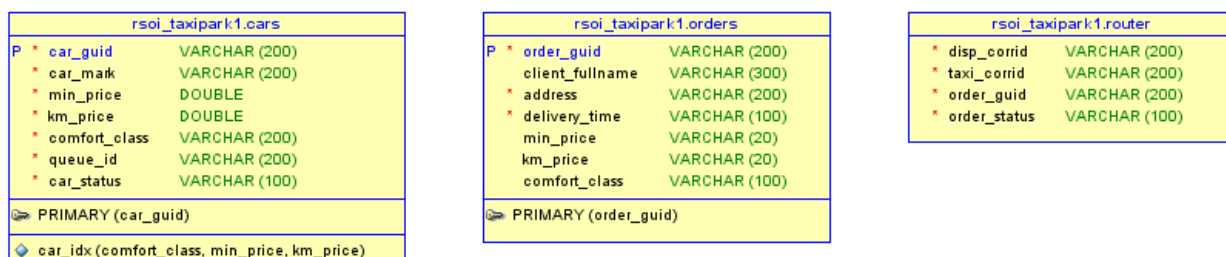


Рисунок 15. Схема БД АИС Таксопарка

Таблица CARS содержит все зарегистрированные в таксопарке автомобили такси, их идентификаторы, а также информацию о них для возможности осуществления поиска подходящих такси на основе фильтра, заданного пользователем.

В таблице ORDERS регистрируются заказы такси.

Таблица ROUTER является маршрутизатором между диспетчерской системой и АИС Такси.

3.1.3 АИС Такси

На рисунке 16 представлена схема БД для АИС Такси:

rsoi_taxi1_taxipark1.car		
P	* car_guid	VARCHAR (200)
	* coordinates	VARCHAR (200)
	* status	VARCHAR (100)
PRIMARY (car_guid)		
car_idx (status)		

Рисунок 16. Схема БД для АИС Такси

В таблице CAR хранится идентификатор автомобиля, текущий статус такси и его координаты.

3.2 Язык программирования и среда разработки

Для разработки данного проекта была платформа Java.

Java – универсальный кросс-платформенный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems. В настоящий момент собственником технологии является компания Oracle Corporation.

Основные преимущества платформы Java:

- возможность запуска приложений под управлением большинства современных операционных систем;
- высокая надежность и безопасность;
- переносимость;
- высокая производительность;
- автоматическое управление выделением памяти;
- простые и удобные возможности разработки web-приложений и распределенных сетевых приложений;
- средства создания многопоточных приложений;
- отсутствие жесткой зависимости от поставщика программного обеспечения.

Для разработки данного проекта была выбрана и среда разработки Eclipse Juno. Она зарекомендовала себя своей стабильностью и широкими возможностями. Также она известна своим широким набором интегрированных инструментов для рефакторинга, которые позволяют программистам быстро реорганизовывать исходные тексты программ. Дизайн среды ориентирован на продуктивность работы программистов, позволяя им сконцентрироваться на разработке функциональности.

3.3 Протоколы взаимодействия

3.3.1 Протокол синхронного взаимодействия

В качестве протокола синхронного взаимодействия был выбран SOAP поверх HTTP. SOAP является открытым стандартом и позволяет реализовывать клиента и сервер на различных языках программирования. Кроме того, SOAP имеет реализации под множество языков программирования, что облегчает написание клиентов. В отличие от XML-RPC SOAP позволяет производить не только удаленные вызовы процедур с примитивными типами, но передавать сложные структуры данных, которые описываются согласно стандарту SOAP и преобразовываются в объекты целевого языка программирования. В отличие же от REST сервисов, SOAP предоставляет WSDL документ, который описывает функции сервиса и форматы передаваемых сообщений, что освобождает разработчика от написания клиента сервиса. Данная задача возлагается на соответствующий генератор кода.

Нижестоящий протокол HTTP был выбран по причине его широкого распространения и возможности использования SSL для шифрования сообщений.

При необходимости шифрования не только на уровне нижестоящего протокола, но и на уровне сообщений, SOAP предоставляет соответствующий стандарт для решения этой задачи.

3.3.2 Протокол асинхронного взаимодействия

В качестве транспортной системы для доставки асинхронных сообщений был выбран продукт компании Apache – ActiveMQ.

Данный продукт является открытым и распространяется под лицензией Apache, что позволяет использовать его для своих нужд.

Основным конкурентом ActiveMQ является продукт компании IBM, который превосходит его по ряду характеристик, однако требует приобретения лицензии.

ActiveMQ предоставляет клиенты для наиболее распространенных языков программирования, а также поддерживает множество коннекторов для взаимодействия. Имеется также коннектор для XMPP.

В отличие от почтового или XMPP сервера ActiveMQ ориентирован на доставку сообщений между АИС, что подразумевает под собой следующие преимущества:

- высокая производительность;
- гарантированная доставка сообщений;
- гарантированный порядок сообщений;
- буферизация;
- транзакционная отправка и прием сообщений;
- поддержка механизма двухфазной фиксации;
- кластеризация транспортных узлов для обеспечения высокой доступности.

3.4 Система сборки

Для сборки проектов на Java, как правило, используется одна из двух систем: Ant или Maven. Java является кросс-платформенным языком, поэтому концептуально не может использовать платформенно-зависимые утилиты, такие как Make.

В данном проекте используется система сборки Maven, так как она является декларативной в отличие от Ant, позволяет описывать всю структуру проекта в одном xml-файле, интегрируется с основными IDE и позволяет управлять зависимостями.

Проект сборки имеет следующие цели:

- `clean` – очистка проекта (удаляет бинарные файлы и остальной «мусор»)
- `compile` – компиляция проекта
- `test` – запуск тестов
- `package` – упаковка файлов в соответствующие архивы для последующей транспортировки и развертывания
- `install` – отправка архивов в удаленный бинарный репозиторий
- `deploy` – разворачивание архивов на сервере
- `tomcat7:run` – запуск встроенного сервера с установленным приложением
- `war:war` – экспорт приложения в виде архива `*.war` для дальнейшей его установки на сервер приложений.
- Безопасная передача данных через SOAP и JMS

3.5 Сервер приложений

В качестве сервера приложений был выбран Apache Tomcat 7. На самом деле он не является полноценным сервером приложений в концепциях Java. Tomcat является лишь контейнером сервлетов, а это означает, что он позволяет развертывать Web-приложения и Web-сервисы на Java. Также Tomcat имеет встроенный Web-сервер, однако при необходимости можно сконфигурировать выделенный Web-сервер для раздачи статического контента.

3.6 База данных

В качестве базы данных для разработки используется БД MySQL. MySQL портирована на большое количество платформ, а также имеет API для многих языков, включая Delphi, C, C++, Java, Perl, PHP, Python, Ruby.

3.7 Развертывание системы

На Рисунке 17 представлена диаграмма развертывания системы заказа такси. Клиенты взаимодействуют с системой через Web-браузер. Взаимодействие с другими системами происходит через SOAP поверх HTTP. Асинхронные запросы выполняются через транспортный сервер, развернутый на отдельном узле. Само приложение системы бронирования выполняется в контейнере сервлетов Apache Tomcat 7 и общается с базой через JDBC.

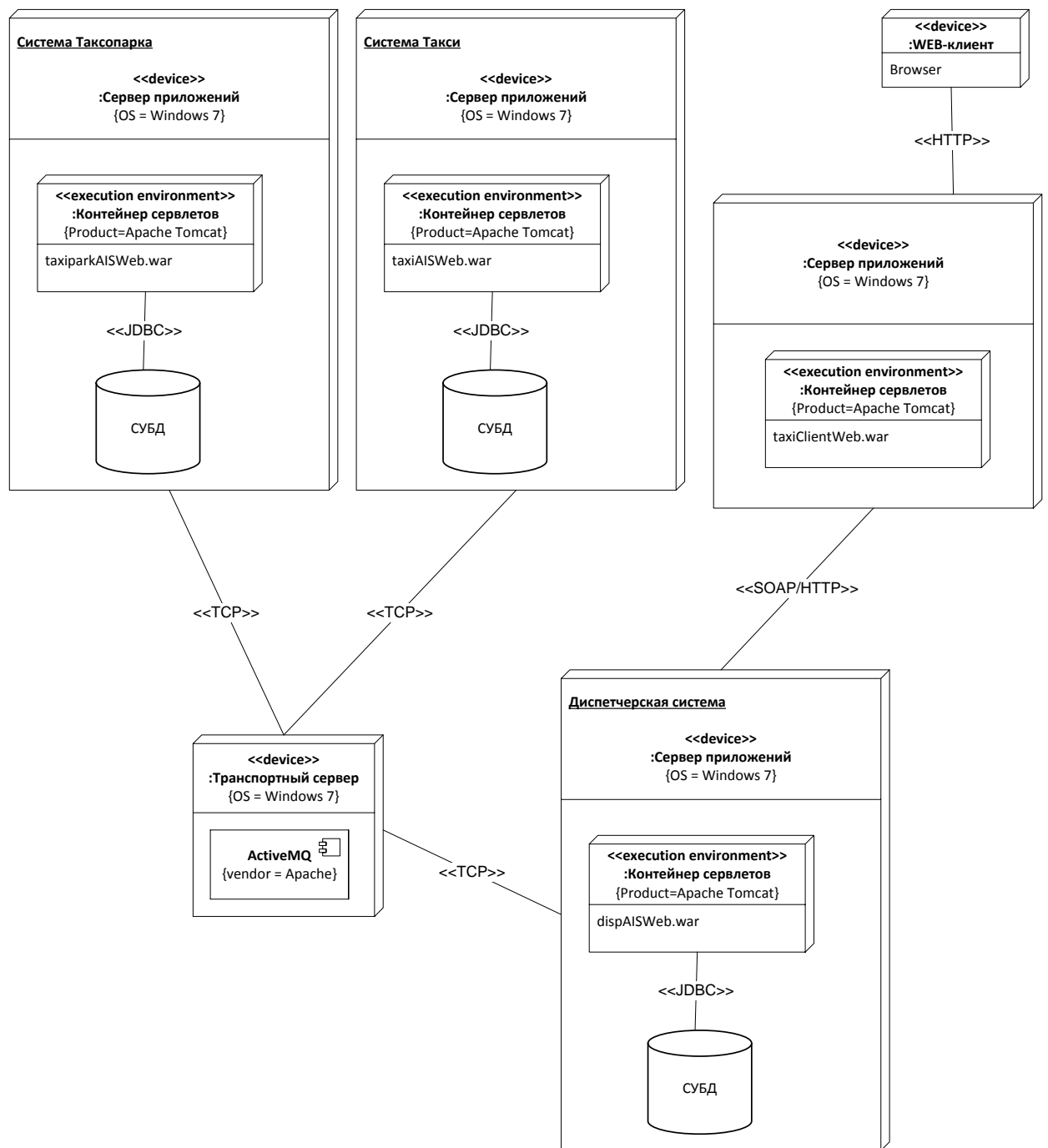


Рисунок 17. Диаграмма развертывания системы заказа такси

3.8 Кэширование

Кэширование реализовано для «Диспетчерской АИС». После подтверждения заказа такси от таксопарка система запускает задачу кэширования статуса заказа, опрашивая «АИС Такси» через заданные промежутки времени и сохраняя текущий статус заказа в локальной БД.

При поступлении запрос от пользователя на получение статуса заказа, «Диспетчерская АИС» возвращает заэкшированное значение статуса из локальной БД.

3.9 Интерфейс

Взаимодействие пользователя с системой заказа такси осуществляется по средствам WEB-интерфейса. На рисунке 18 представлена главная страница системы.

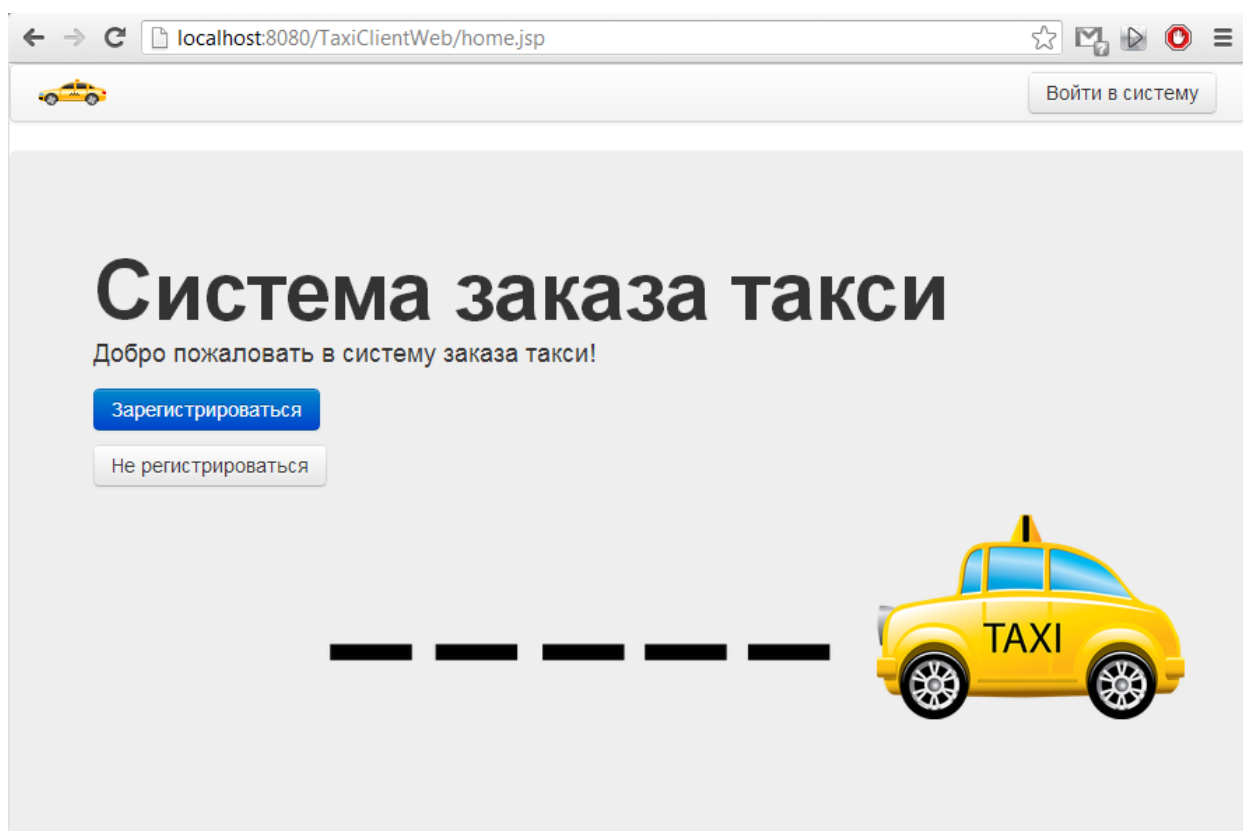


Рисунок 18. Главная страница системы заказа такси

С главной страницы системы пользователь может выполнить регистрацию в системе или вход в систему – рисунках 19 и 20, соответственно:

The image shows a registration form window titled "Регистрация" (Registration) with a close button (X) in the top right corner. The form contains five input fields: "Имя" (Name), "Фамилия" (Surname), "Отчество" (Patronymic), "Логин" (Login), and "Пароль" (Password). Each field is followed by a red label "Обязательно" (Mandatory). At the bottom right, there are two buttons: a grey "Закрыть" (Close) button and a blue "Регистрация" (Registration) button.

Рисунок 19. Форма регистрации пользователя в системе

The image shows a login form window titled "Вход в систему" (Login to the system) with a close button (X) in the top right corner. The form contains two input fields: "Логин" (Login) and "Пароль" (Password). Each field is followed by a red label "Обязательно" (Mandatory). At the bottom right, there are two buttons: a grey "Закрыть" (Close) button and a blue "Войти" (Login) button.

Рисунок 20. Форма входа пользователя в систему

После успешной авторизации в системе пользователь перенаправляется на домашнюю страницу системы заказа такси. Домашняя страница представлена на рисунке 21.

Рисунок 21. Домашняя страница системы заказа такси

На рисунке 22 представлена форма просмотра истории заказов пользователя:

Адрес	Время доставки	Цена за км.	Цена за мин.	Класс комфорта	Время заказа	Статус заказа
krondshtadsk, 43, k1, kv 25	2013-05-21T01:13:31+04:00	0.0	0.0	free	2013-05-21T01:11:05+04:00	free
krondshtadsk, 43, k1, kv 25	2013-05-21T01:09:06+04:00	88.0	99.0	free	2013-05-21T01:06:19+04:00	free
krondshtadsk, 43, k1, kv 25	2013-05-23T07:08:00+04:00	77.0	67.0	UNKNOWN	2013-05-21T00:45:04+04:00	UNKNOWN

Рисунок 22. Форма просмотра истории заказов пользователя

В случае возникновения ошибок при работе в системе, пользователь перенаправляется на страницу с ошибкой. Пример страницы с ошибкой изображен на рисунке 23:

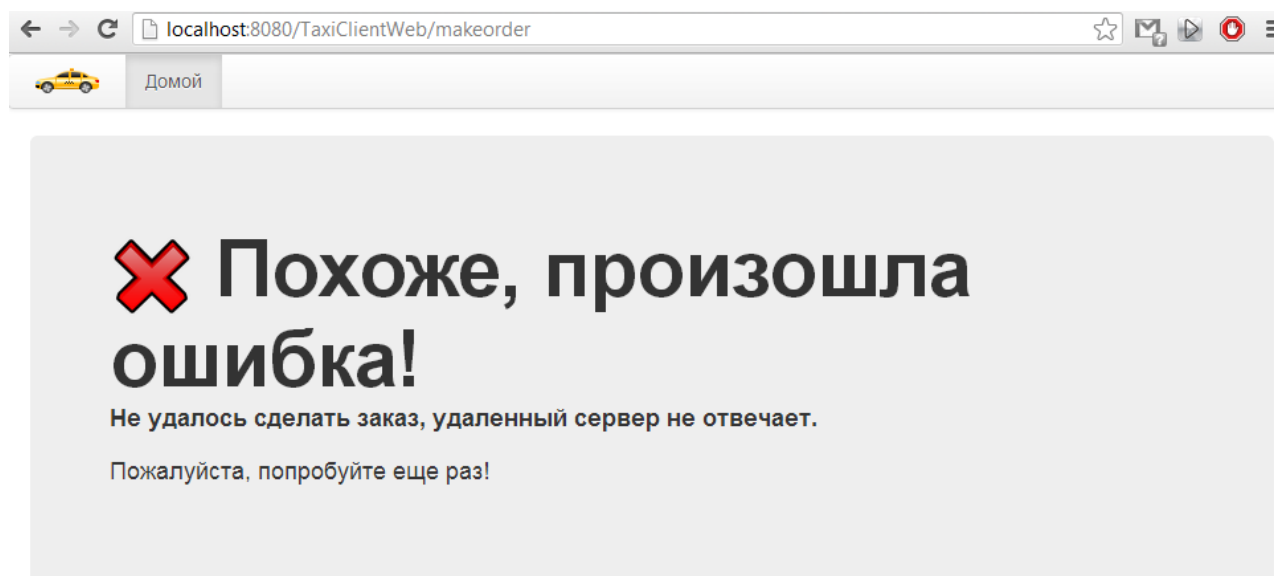


Рисунок 23. Страница с ошибкой

3.10 Используемые библиотеки

- Spring
- Log4j
- JUnit
- ActiveMQ

3.11 Тестирование системы

В ходе разработки системы заказа такси было проведено модульное и интеграционное тестирование.

Для модульного и интеграционного тестирования использовалась библиотека JUnit и система сборки Maven для запуска тестов.

3.11.1 Модульное тестирование

В таблицах 9, 10, 11 приведены результаты тестирования Диспетчерской АИС, АИС Таксопарка и АИС Такси соответственно.

Таблица 9. Модульные тесты Диспетчерской АИС.

№	Описание теста	Входные данные	Ожидаемый результат	Полученный результат
1	Попытка входа в систему с некорректными логином / паролем	Логин пользователя, пароль пользователя	Отказ в авторизации	Отказ в авторизации
2	Попытка входа с правильными логином / паролем	Логин пользователя, пароль пользователя	Успешная авторизация	Успешная авторизация
3	Заявка на заказ такси с указанием прошедшей даты	Идентификатор пользователя, параметры заказа	Отказ в оформлении заявки	Отказ в оформлении заявки
4	Заявка на заказ такси с правильно заполненными параметрами	Идентификатор пользователя, параметры заказа	Сообщение об успешном принятии заказа в обработку	Сообщение об успешном принятии заказа в обработку
5	Получение статуса заказа для незарегистрированного пользователя, с указанием некорректного кода доступа	Код доступа незарегистрированного пользователя	Сообщение о невозможности получить статус заказа	Сообщение о невозможности получить статус заказа
6	Получение статуса заказа для	Код доступа незарегистрированного	Статус текущего заказа	Статус текущего заказа

	незарегистрированно го пользователя, с указанием корректного кода доступа	го пользователя	незарегистрированно го пользователя	незарегистрированно го пользователя
--	---	-----------------	--	--

Таблица 10. Модульные тесты АИС Таксопарка.

№	Описание теста	Входные данные	Ожидаемый результат	Полученный результат
1	Заявка на получение возможности заказа с некорректными полями сообщения	Сообщение на получение возможности заказа	Сообщение об ошибке	Сообщение об ошибке
2	Заявка на получение возможности заказа с корректными полями сообщения	Сообщение на получение возможности заказа	Сообщение, содержащее статус	Сообщение, содержащее статус
3	Заявка на подтверждение заказа с некорректным идентификатором заказа	Идентификатор заказа	Сообщение о невозможности подтверждения заказа	Сообщение о невозможности подтверждения заказа
4	Заявка на подтверждение заказа с корректным идентификатором заказа	Идентификатор заказа	Сообщение об успешном принятии заказа в обработку	Сообщение об успешном принятии заказа в обработку

Таблица 11. Модульные тесты АИС Такси.

№	Описание теста	Входные данные	Ожидаемый результат	Полученный результат
1	Заявка на получение статуса такси с	Сообщение на получение статуса	Сообщение об ошибке	Сообщение об ошибке

	некорректными полями сообщения	такси		
2	Заявка на получение статуса такси с корректными полями сообщения	Сообщение на получение статуса такси	Сообщение, содержащее статус	Сообщение, содержащее статус
3	Заявка на подтверждение заказа с некорректными полями сообщения	Сообщение на подтверждение заказа	Сообщение о невозможности подтверждения заказа	Сообщение о невозможности подтверждения заказа
4	Заявка на подтверждение заказа с некорректными полями сообщения	Сообщение на подтверждение заказа	Сообщение об успешном принятии заказа в обработку	Сообщение об успешном принятии заказа в обработку

3.11.2 Системное тестирование

В таблице 12 приведены системные тесты для системы заказа такси.

Таблица 12. Системные тесты системы заказа такси.

№	Описание теста	Входные данные	Ожидаемый результат	Полученный результат
1	Попытка одновременного заказа 1го автомобиля такси 2мя клиентами	Идентификатор клиента, параметры заказа	Отказ в заказе для одного из клиентов. Подтверждение для другого клиента.	Отказ в заказе для одного из клиентов. Подтверждение для другого клиента
2	Попытка одновременного заказа такси несколькими клиентами	Идентификатор клиента, параметры заказа	Успешная обработка заказов от всех клиентов в параллельном режиме	Успешная обработка заказов от всех клиентов в параллельном режиме
3	Поступление заказов	Идентификатор	Постановка задачи	Постановка задачи

	от пользователей, чье количество превышает число доступных автомобилей	клиента, параметры заказа	всем таксистам. Часть заказов переведены в ожидание	всем таксистам. Часть заказов переведены в ожидание
--	--	------------------------------	--	--

Заключение

Результатом курсовой работы является закрепление навыков описания процессов автоматизируемой предметной области, проектирования и программной реализации автоматизированных систем, использование системы контроля версий, средств модульного и интеграционного тестирования, подготовки программной документации и тестирования разработанного ПО.

В ходе работы над курсовым проектом была разработана распределенная система заказа такси, позволяющая клиентам осуществлять заказ такси с использованием различных фильтров.

Разработанная система состоит из 3 подсистем: диспетчерская система, система таксопарков, система такси.

Список использованных источников

1. Крищенко В. А. Конспект лекций по курсу «Распределенные системы обработки информации».
2. Крищенко В. А. Технологии создания кросс-платформенных распределённых приложений. Учебное пособие.
3. В. Basham, К. Sierra, В. Bates. Head First Servlets and JSP.
4. М. Kalin. Java Web Services: Up and Running.
5. <http://activemq.apache.org/> ActiveMQ
6. www.springsource.org/ Spring