МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

КУРСОВАЯ РАБОТА   
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | А. Э. Зянчурин |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗИРОВАННОЙ/ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОБУСНОГО ВОКЗАЛА |
| по дисциплине: МЕТОДОЛОГИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4232М |  |  |  | В. Ф. Губайдулин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

СОДЕРЖАНИЕ

[ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗИРОВАННОЙ/ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОБУСНОГО ВОКЗАЛА 1](#_Toc135684018)

[1 Введение 5](#_Toc135684019)

[1.1 Актуальность 5](#_Toc135684020)

[1.2 Цель 6](#_Toc135684021)

[1.3 Объект и предмет 6](#_Toc135684022)

[1.4 Новизна технических решений 7](#_Toc135684023)

[1.5 Структура и объём работы 7](#_Toc135684024)

[2 Назначение и область применения 8](#_Toc135684025)

[2.1 Системный анализ предметной области и методов решения задачи автоматизации 8](#_Toc135684026)

[2.2 Краткая характеристика условий и области применения изделия 9](#_Toc135684027)

[2.3 Назначение изделия, постановка задачи автоматизации и цифровизации предметной области 10](#_Toc135684028)

[3 Техническая характеристика 15](#_Toc135684029)

[3.1 Анализ требований технического задания 15](#_Toc135684030)

[3.2 Описание вида автоматизируемой деятельности. Критерии и характеристики качества изделия. Предложения по совершенствованию 17](#_Toc135684031)

[3.3.1 Детализация и структуризация целей разрабатываемого изделия 18](#_Toc135684032)

[3.3.2 Определение класса изделия как информационной системы 19](#_Toc135684033)

[3.3.3 Разработка контекстной диаграммы 19](#_Toc135684034)

[3.3.4 Анализ соответствия концептуального описания и требований ТЗ 20](#_Toc135684035)

[3.3.5 Принцип построения изделия 21](#_Toc135684036)

[3.4 Сведения о соответствии или отклонениях от требований, установленных техническим заданием с обоснованием отклонения 21](#_Toc135684037)

[4 Описание и обоснование выбранной конструкции 23](#_Toc135684038)

[4.1 Моделирование предметной области 23](#_Toc135684039)

[4.2 Функциональная структура изделия. Алгоритмизация решения задачи автоматизации процессов предметной области 27](#_Toc135684040)

[4.3 Моделирование программной структуры и архитектурное описание изделия 29](#_Toc135684041)

[4.4 Сравнительный анализ аналогов изделия 31](#_Toc135684042)

[4.5 Перспективы применения технологических и технических решений 31](#_Toc135684043)

[5 Описание организации работ с применением разрабатываемого изделия 33](#_Toc135684044)

[5.1 Организация коллективной разработки 33](#_Toc135684045)

[5.1.1 Ролевая модель команды 33](#_Toc135684046)

[5.1.2 Правила выполнения разработки 33](#_Toc135684047)

[5.2 Развертывание и поставка программного обеспечения 35](#_Toc135684048)

[Приложение А. Перечень работ 36](#_Toc135684049)

[Приложение Б. Формализованные функциональные требования 38](#_Toc135684050)

[Приложение В. Структура данных 43](#_Toc135684051)

[Приложение Г. Уточнение сетевого плана графика 47](#_Toc135684052)

[Перечень литературы и документов, используемых при разработке пояснительной записке к техническому проекту 48](#_Toc135684053)

[Вывод 49](#_Toc135684054)

[Список используемой литературы 50](#_Toc135684055)

**Программное обеспечение** **для автоматизации/информационной системы автобусного вокзала**

**Технический проект**

*technicaldocs.ru*

*Подпись и дата*

*Взам. инв. №*

*Инв. № дубл.*

*Инв. № подл.*

*Подпись и дата*

# 1 Введение

# 1.1 Актуальность

Разрабатываемая система призвана автоматизировать большую часть процессов, протекающих на автовокзале, такие как: покупка билетов, нотификация и индикация актуальных рейсов, внутреннее управление и распределение рабочих ресурсов: транспортные средства (автобусы) и водители.

На момент начала разработки вышеперечисленные процессы протекают следующим образом [1]:

* покупка билетов – производится исключительно с помощью контроллеров-кассиров. Покупатель не видит все рейсы, оставшееся количество билетов, реальную цену. Есть вероятность, что контроллеры могут сохранять последние билеты специально, с целью продажи их своим знакомым, либо могут завышать цену. Контроль за контроллерами отсутствует. Также, имеет место быть проблема формирования длинных людских очередей у контроллеров.
* нотификация и индикация актуальных рейсов – производится исключительно с помощью громкого голосового объявления, отсутствует визуальное отображение рейсов. Присутствует проблема мискоммуникации, не редки случаи, когда пассажир опаздывал на свой рейс или задерживал его из-за того, что не смог вовремя услышать голосовое объявление.
* распределение рабочих ресурсов – водители и транспортные средства распределяются с помощью ненастроенной модификации 1С: Предприятие 8. Не настроен учёт работы водителей, лимит амортизации транспортного средства. Приказ Министерства транспорта РФ от 16 октября 2020 г. № 424 “Об утверждении Особенностей режима рабочего времени и времени отдыха, условий труда водителей автомобилей” исполняется посредственно.

Реализация процессов с помощью разрабатываемой системы: [2]

* покупка билетов – появляется возможность приобрести билеты с помощью интерактивных терминалов. У контроллеров-кассиров пропадает возможность держать билеты для знакомых или производить продажу по завышенной цене. Решается проблема с формирование огромных очередей.
* нотификация и индикация актуальных рейсов – вдобавок к голосовым объявлениям будет реализовано табло актуальных рейсов, которое обеспечит визуальное донесение актуальной информации, что значительно снизит количество случаев, когда пассажир опаздывал на рейс.
* распределение рабочих ресурсов – будет реализовано специальное АРМ Оператора, которое позволит чётко производить учёт отработанного времени водителем. АРМ будет интуитивно понятно и удобно в использовании, в отличии от перегруженного и избыточного 1С.

# 1.2 Цель

Целью технического проектирование является повышение эффективности работы предприятия, значительное улучшение опыта получения услуг для пассажиров, упрощение и гарантирование верного учёта ресурсов (водители, автобусы).

# 1.3 Объект и предмет

Объект – web-приложение, включающее в себя функционал покупки билетов, отображения рейсов, АРМ оператора.

Предмет – проектирование на основе собранных требований, реализация спроектированного решения, дальнейшая поддержка готовой системы.

# 1.4 Новизна технических решений

Решение включает в себя типовые функции (покупка, отображение, управление), которые необходимо адаптировать под конкретную область применения – автовокзал, а также под конкретные условия использования. Система включает в себя две части: backend и frontend, которые будут запущены отдельно друг от друга, но будут работать исключительно в связке друг с другом. [15]

# 1.5 Структура и объём работы

Количество приложений – 4.

Количество литературных источников – 2.

Количество рисунков – 14.

Количество таблиц – 2.

Список сокращений:

АРМ – автоматизированное рабочее место

ТС – транспортное средство

БД – база данных

# 2 Назначение и область применения

# 2.1 Системный анализ предметной области и методов решения задачи автоматизации

В текущих процессах в уже имеющейся системе управления автовокзалом имеют место быть недочёты, упущения, а самое главное нарушения касающихся учёта ресурсов. Разрабатываемая система будет призвана комплексно подойди к решению и исправлению всех нестыковок, чтобы автоматизировать протекающие процессы, а также улучшить опыт пассажиров, которые прибегают к услугам данного автовокзала.

Все процессы в системе имеют влияние на различных стадиях выполнения.

Двумя основами системы являются ресурсы (водители и автобусы) и рейсы. Для грамотного учёта ресурсов необходима краткая и не конфиденциальная информация о каждом ресурсе конкретно. Для водителей это ФИО, отработанное время, а также факт доступности водителя. Для автобусов важной информацией являются регистрационный номер, количество мест и состояние транспортного средства. Исходя из этих данных появляется возможность формирования рейсов. Рейс формируется за счёт информации о дате и времени начала рейса, дате и времени окончания, а также о водители и автобусе, обслуживающих этот рейс. Также, важным атрибутом рейса является количество билетов, доступных для покупки. Данный атрибут позволяет осуществлять продажу билетов на рейс, и ограничивает случаи, когда продаётся билетов более, чем нужно или возможно. После формирования рейса, предоставляется возможность купить билет на сформированный рейс. После покупки билета, необходимо осуществлять должное информирование, для указания действительно качественной услуги. [4, 6]

Исходя из вышеописанных данных модель можно разделить на 3 части, которые зависимы друг от друга. Зависимы они, так как без формирования рейса невозможно приобрести билет на рейс, а без приобретения билета, у пассажира нет надобности следить за обновлениями рейса. Части модели представлены на рисунке 1.

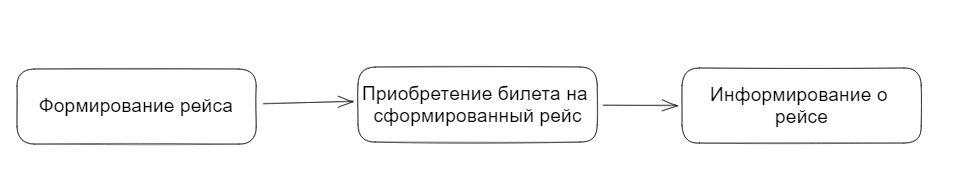


Рисунок 1 – Упрощённая модель процессов на автовокзале

Для формирования рейса появляется необходимость в реализации АРМ оператора, который будет предоставлять возможность быстро и удобно вносить изменения в имеющиеся рейсы, а также ресурсы. Для приобретения билетов необходимо реализовать интерактивный интерфейс для покупки билетов. Информирование необходимо производить с помощью визуальных инструментов, список актуальных рейсов. [2, 4]

Решение будет представлять из себя web-приложение на основе Rest API. Система будет защищена с помощью авторизации, а доступ к частям модели будет осуществляться исходя из ролей.

# 2.2 Краткая характеристика условий и области применения изделия

Интересантами реализуемой системы будут операторы, а также пассажиры. Операторам будет доступен АРМ оператора для назначения ТС и водителей на рейс, которое позволит операторам обеспечивать рейсы, а также оперативно реагировать на изменения, связанные с различными форс-мажорами (болезнь водителя, поломка ТС). [2, 5] Пассажиры смогу напрямую пользоваться интерактивными терминалами для покупки билетов, которые предоставляют покупателям возможности выбора подходящего рейса на выбранную дату, выбора количества билетов, оплаты безналичным способом, и также косвенно пользоваться табло с рейсами, которое предоставит пассажирам актуальную информацию и ближайших рейсах. [7] Таким образом получается, что все процессы являются существенными для рассмотрения.

Сопровождение необходимо разделить на 2 части: поддержка оборудования (интерактивные терминалы, табло, компьютеры операторов, серверы), поддержка непосредственно программы. [15] За поддержку оборудования несут ответственность инженеры заказчика. За поддержку программы несёт ответственность подрядчик, разрабатывающий решение.

Приемка результатов осуществляется на средствах вычислительной техники Заказчика в объеме, определяемом программой и методикой испытаний, и в сроки, определенные ведомостью исполнения (календарным планом) работ к соответствующему договору. [15] Результаты приемки должны быть отражены в техническом акте. Порядок приёмки представлен в приложении А. Оценка результатов будет производится на основе выполнения сценария тестирования, который будет составлен тестировщиками подрядчика. Инженерами по производительности подрядчика следует составить и выполнить план нагрузочного тестирования, результаты оформить в виде отчёта.

Разрабатываемое приложение будет развёрнуто в частной сети автовокзала, все сервера в данной сети имеют белые (статичные) IP адреса. Доступ к API или интерфейсам будет невозможно получить, если устройство клиента находится за внутренней сетью. [8, 5] В системе будет отсутствовать ручной режим из-за ненадобности, чтобы не усложнять систему. После развёртывания система уже находится в рабочем состоянии, для полной работоспособности остаётся только заполнить справочники рейсов, водителей и автобусов через интерфейс АРМ оператора.

Реструктуризация предприятия для реализации системы не является важным вопросом из-за присутствия в системе уже имеющейся позиции оператора, другие позиции не используются.

# 2.3 Назначение изделия, постановка задачи автоматизации и цифровизации предметной области

Функциональное назначение. [2]

Автоматизация следующих процессов на автобусных вокзалах:

* самостоятельная покупка билетов пассажирами с использованием специальных интерактивных терминалов;
* информирование об актуальных рейсах путём визуализации списка рейсов на табло;
* назначение ТС и водителей на рейсы путём реализации АРМ оператора.

Эксплуатационное назначение. [2]

Автоматизация и интеграция процессов с использованием:

* интерактивных терминалов для покупки билетов. Терминалы будут располагаться на территории автовокзала в количестве от 10 до 15 штук. Призваны снизить нагрузку на кассиров, а также оптимизировать время, которое клиенты автовокзала затрачивают на приобретение билетов. Терминалы предоставляют покупателям возможности выбора подходящего рейса на выбранную дату, выбора количества билетов, оплаты безналичным способом.
* табло с рейсами, которое предоставит пассажирам актуальную информацию и ближайших рейсах.
* АРМ оператора для назначения ТС и водителей на рейс, которое позволит операторам обеспечивать рейсы, а также оперативно реагировать на изменения, связанные с различными форс-мажорами (болезнь водителя, поломка ТС).

Был определён список основных функций.

Список основных функций:

1. Покупка билета
2. Фильтрация списка билетов по дате
3. Фильтрация списка билетов по конечному пункту
4. Выбор количества билетов на выбранный рейс
5. Оплата по QR-коду
6. Демонстрация актуальных рейсов
7. Назначение ТС и водителя на рейс
8. Фильтрация рейсов по дате
9. Генерация билетов
10. Отмена рейса
11. Редактирование списка ТС
12. Удаление ТС
13. Добавление ТС
14. Изменение имеющихся ТС
15. Редактирование списка водителей
16. Удаление водителей
17. Добавление водителей
18. Изменение имеющихся водителей
19. Редактирование ЛК операторов
20. Удаление оператора
21. Добавление оператора
22. Изменение имеющихся операторов
23. Редактирование шаблонов рейсов
24. Удаление шаблона рейсов
25. Добавление шаблона рейсов
26. Изменение имеющихся шаблонов рейсов

Было сформировано дерево функций изделия, которое представлено на рисунке 2.

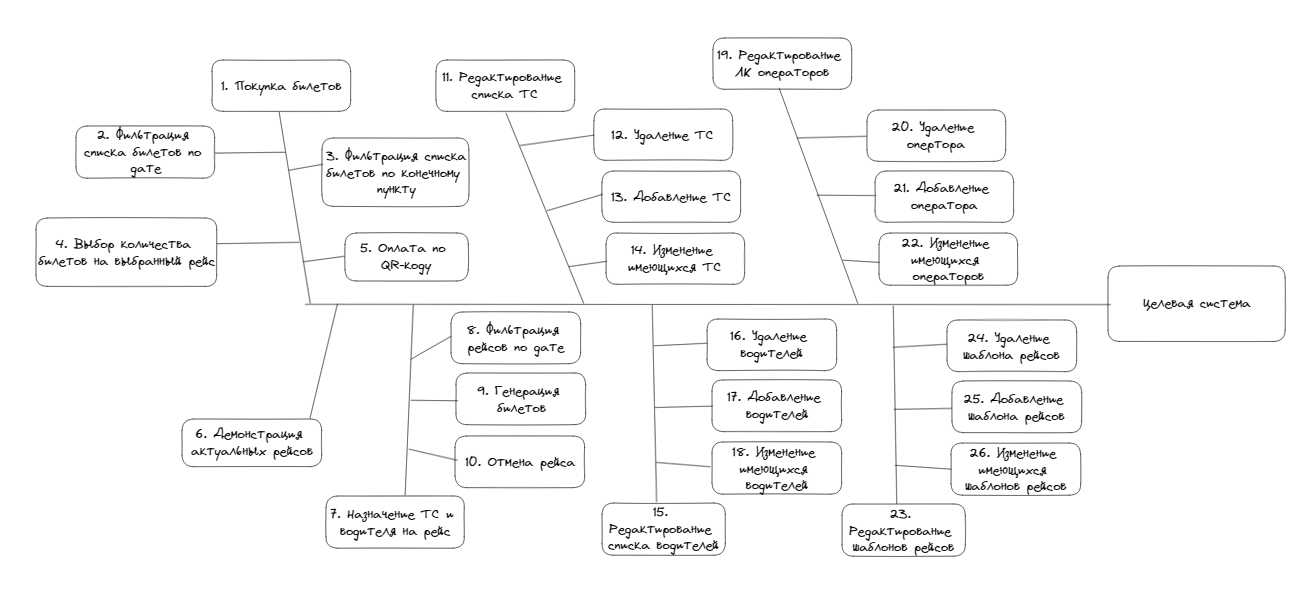


Рисунок 2 – Дерево функций изделия

По результатам проведённого анализа установлено, что для повышения эффективности целесообразно внедрение системы управления автовокзалом. Данная система сможет значительно повысить эффективность всех процессов, протекающих на автовокзале, а также исключить или убрать серьёзные нарушения и уязвимости. Таким образом, представляется актуальным решение следующих задач:

1. Анализ требований ТЗ на систему управление автовокзалом;

2. Разработка моделей, связанных с процессами;

3. Разработка архитектуры системы;

4. Сравнительное исследование эффективности применения с аналогами;

5. Испытание написанного кода системы;

6. Написание технической документации системы;

7. Запуск системы;

8. Эксплуатационное тестирование системы;

9. Окончательная передача разработанной и готов системы;

10. Поддержка системы и работы по актуализации.

# 3 Техническая характеристика

# 3.1 Анализ требований технического задания

После проведённого анализа требований не было обнаружено конфликтов между требованиями, их разрешение не необходимо.

Назначение программного обеспечения для автоматизации/информационной системы автобусного вокзала заключается в автоматизации следующих процессов:

* покупка билетов;
* демонстрация актуальных рейсов;
* назначение водителей и ТС на рейсы.

Остальные процессы автобусного вокзала не включены в реализацию системы автоматизации.

Вся система будет реализована на веб-технологиях: Django + Angular.

Покупка билетов будет производится через интерактивный сенсорный терминал, на котором будет запущена страница для покупки билетов. Клиенту автобусного вокзала будет дана возможность выбрать интересующий его рейс, выбрать количество билетов, которое он хочет приобрести, а также оплатить заказ по QR-коду.

Демонстрация актуальных рейсов будет производится с помощью электронного табло (монитора), на котором будет запущена страница с таблицей рейсов. Раз в *n* минут таблица будет обновляться, отсекая неактуальные заказы.

АРМ оператора будет запущен на ПК оператора. АРМ позволяет назначать водителей и ТС на рейсы с помощью технологии Drag&Drop. В зависимости от выбранного транспорта, будут генерироваться билеты на назначенный рейс.

Требования к интерактивным терминалам по продаже билетов: [1]

* возможность фильтрации рейсов;
* демонстрация актуальной информации о рейсе: стоимость билета, количество оставшихся мест, время в пути, тип ТС (марка);
* возможность выбрать конкретный билет и их количество;
* оплата по QR-коду.

Требования к информационному табло: [1]

* демонстрация актуальной информации о рейсе: время отправления, номер остановки;
* демонстрация только будущих рейсов, относительно текущих даты и времени.

Требования к АРМ оператора: [1]

* назначение ТС и водителя с использованием технологии drag&drop;
* создание билетов на основании вместительности выбранного ТС;
* возможность изменить назначенное ТС, если новое ТС содержит мест не меньше, чем уже было продано билетов;
* возможность назначения нескольких автобусов на рейс;
* возможность назначить только того водителя, у которого нет переизбытка рабочего времени.

Общие требования к системе: [1]

* наличие строгой ролевой модели: покупатель, табло, оператор, администратор;
* наличие индивидуальных аккаунтов у операторов;

Все формализованные функциональные требования представлены в приложении Б.

Все вышеописанные требования являются осуществимыми.

В техническом задании и спецификации требований были классифицированы, декомпозированы и проанализированы:

* Функциональные требования:
  + Бизнес-требования
  + Системные требования
* Нефункциональные требования:
  + Атрибуты качества
  + Внешние интерфейсы

Была подготовлена следующая приоретизация требований: [3]

1. Бизнес требования

2. Атрибуты качества

3. Системные требования

4. Внешние интерфейсы

# 3.2 Описание вида автоматизируемой деятельности. Критерии и характеристики качества изделия. Предложения по совершенствованию

Разрабатываемая система имеет явно выраженную деятельность автоматизированной системы управления (АСУ ТП). Данная система позволяет работать с информацией о рейсах, водителях и автобусах, позволяется её создавать, изменять или удалять. [5, 12]

Для получения характеристик и критериев будущего изделия была составлена таблица 1, в которой продемонстрированы функции, критерии и характеристики изделия.

Таблицы 1 – Функции, критерии и характеристики изделия

|  |  |  |
| --- | --- | --- |
| **Функция** | **Характеристика** | **Критерии** |
| Покупка билетов | 1. Скорость выполнения  2. Грамотный UX/UI  3. Актуальность | 1. Сохранение корректности данных  2. Генерация валидного QR-кода оплаты |
| Демонстрация актуальных рейсов | 1. Актуальность | 1. Демонстрация только актуальной и верной информации |
| Назначение автобуса и водителя на рейс | 1. Предсказуемость | 1. Сохранение корректности данных |
| Редактирование списка ТС | 1. Предсказуемость | 1. Сохранение корректности данных |
| Редактирование списка водителей | 1. Предсказуемость | 1. Сохранение корректности данных |
| Редактирование шаблонов рейсов | 1. Предсказуемость | 1. Сохранение корректности данных |

Дополнения по описанному ТЗ отсутствуют.

**3.3 Концепция и принципы построения изделия. Требования технического задания и их выполнения**

# 3.3.1 Детализация и структуризация целей разрабатываемого изделия

Описание целей разрабатываемого изделия, а также подходы для достижения целей представлены в таблице 2.

Таблица 2 – Описание целей разрабатываемого изделия

|  |  |  |
| --- | --- | --- |
| **Цель** | **Решаемая проблема** | **Что необходимо реализовать** |
| Ввести полный и структурированный учёт ресурсов | Выполнение норм, прописанных для законной работы автобусной станции | Системе необходимо запрещать операторам назначать неподходящие на рейс |
| Улучшить опыт пассажиров в ожидании их рейса | Увеличение репрезентативности информации об актуальных рейсах | Реализовать наглядное и удобное табло актуальных рейсов |
| Оптимизировать очереди у касс, в которых приобретаются билеты | Увеличить количество обсуживающих пассажиров мест, в которых можно приобрести билет на рейс | Реализовать интерактивный интерфейс покупки билетов, который будет доступен пассажирам |
| Увеличить контроль за кассирами | Минимизировать возможность кассиров «задерживать» билеты до нужного покупателя или продавать их по завышенной цене | Реализовать интерактивный интерфейс покупки билетов, который будет доступен пассажирам |

# 3.3.2 Определение класса изделия как информационной системы

По масштабу разрабатываемое изделие можно отнести к групповой системе, за счёт того, что система представляет специальные возможности в зависимости от роли. Для каждой роли предназначен свой интерфейс.

По архитектуре систему смело можно отнести к трёхслойной архитектуре за счёт многоуровневости: уровень СУБД, уровень API (backend) и уровень визуальных интерфейсов (frontend). [11]

По характеру использования информации программа относится к управляющим. Данные представляются в реляционной модели, SQL сервер на движке MS SQL. Система относится к ERPII, за счёт АРМ с помощью которого происходит планирование ресурсов, а также позволяет управлять взаимоотношениями с клиентами-пассажирами. [9] По степени автоматизации система относится к автоматизированным системам, так как роль человека остаётся ключевой.

# 3.3.3 Разработка контекстной диаграммы

На рисунке 3 представлена контекстная диаграмма разрабатываемой системы.

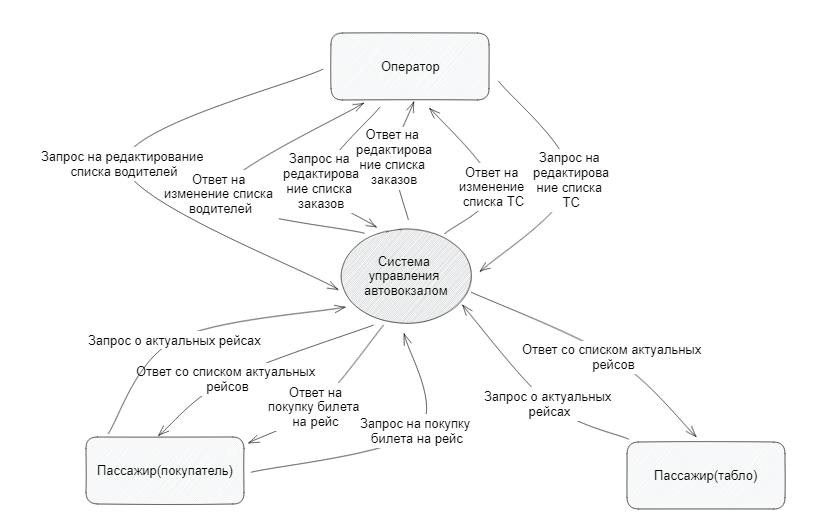


Рисунок 3 – Контекстная диаграмма для разрабатываемой системы

# 3.3.4 Анализ соответствия концептуального описания и требований ТЗ

После проведённого анализа можно с уверенной зафиксировать, что под каждый класс требований ТЗ будет предусмотрен функциональный элемент изделия, такие как редактирование списков водителей, рейсов, ТС, информация об актуальных рейсах и покупка билетов. Класс ИС полностью совпадает с целями разработки, так как система будет позволять планировать ресурсы и управлять взаимоотношениями между клиентам-пассажирами. По всем элементам изделия даны необходимые пояснения о их назначении, чётко обозначены роли, которые будут взаимодействовать с приведёнными функциональными блоками. Полнота требований ТЗ относительно целей и функционала изделия соблюдены в достаточной мере. Все вышеуказанные требования ТЗ осуществимы. Каждое указанное требование является важным для разработки, так как при не реализации одного из блока, технический процесс будет невозможно продолжить. Например, если будет отсутствовать АРМ оператора, невозможно будет осуществлять контроль за билетами на рейсы, так как рейсы будет невозможно обслуживать. Либо же, если не будет реализована покупка билетов, тогда клиенты-пассажиры смогут покупать билеты только у кассиров, что потенциально может привести спекуляциям и другим критическим нарушениям. Также, если не будет реализовано табло, пассажиры всё также будут задерживать, а то и пропускать рейсы, на которые были приобретены билеты.

# 3.3.5 Принцип построения изделия

Ниже представлены принципы построения разрабатываемого изделия:

- операторам стоит отказаться от уже установленной 1С: Предприятия, чтобы не повторять работу в двух местах, а вести всю работу только в web-интерфейсе;

- в качестве хранилища необходимо развернуть SQL сервер на движке MS SQL, ОС – Debian 11. Информация на сервер будет поступать исключительно за счёт миграций технологией ORM, которая заложена в фреймворке Django;

- для развёртывания приложения необходимо выделить сервер с ОС Debian 11, на данном сервере должен присутствовать Docker;

- на территории автовокзала необходимо установить N-ое количество интерактивных сенсорных терминалов, на каждый терминал необходимо установить Samsung Knox для того, чтобы пассажиры видели только web-страницу, и могли перемещать исключительно внутри неё.

# 3.4 Сведения о соответствии или отклонениях от требований, установленных техническим заданием с обоснованием отклонения

Отклонений и соответствий в требованиях не было обнаружено.

# 4 Описание и обоснование выбранной конструкции

# 4.1 Моделирование предметной области

Было принято решение реализовать моделирование предметной области на основе следующих UML диаграмм: диаграмма прецедентов, диаграмма классов, диаграмма последовательности, а также диаграмма состояний. Другие диаграммы считаются избыточными для моделирования системы. [12, 13]

Была реализована диаграмма прецедентов для пользователей-пассажиров, которая представлена на рисунке 4:



Рисунок 4 – Диаграмма прецедентов для пользователей-пассажиров

Была реализована диаграмма прецедентов для операторов, которая представлена на рисунке 5.

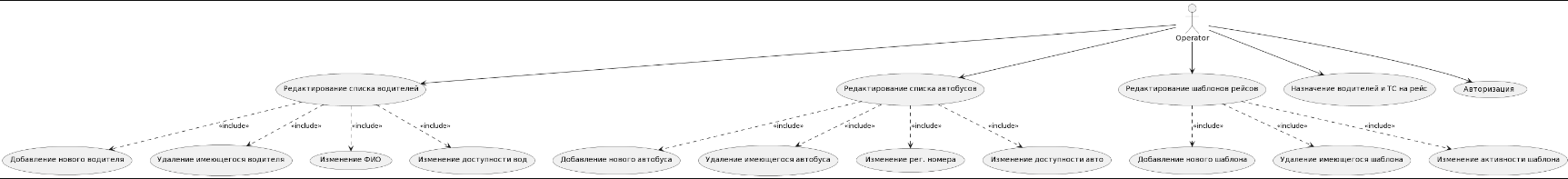


Рисунок 5 – Диаграмма прецедентов для оператора

Была реализована диаграмма классов, которая представлена на рисунке 6.

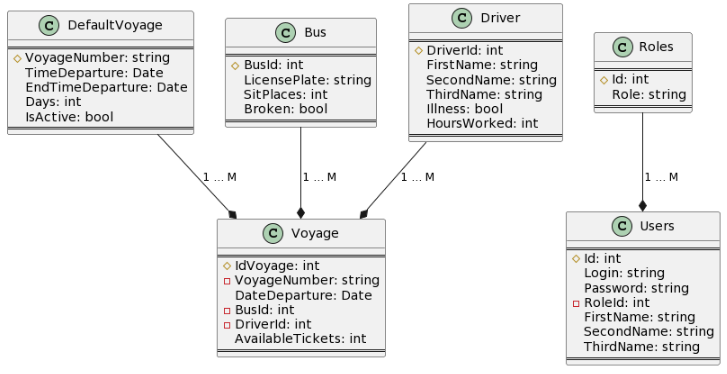


Рисунок 6 – Диаграмма классов

Структура данных представлена в приложении В.

Для действия «Покупка билетов через интерактивный терминал» была реализована диаграмма последовательности, которая представлена на рисунке 7.

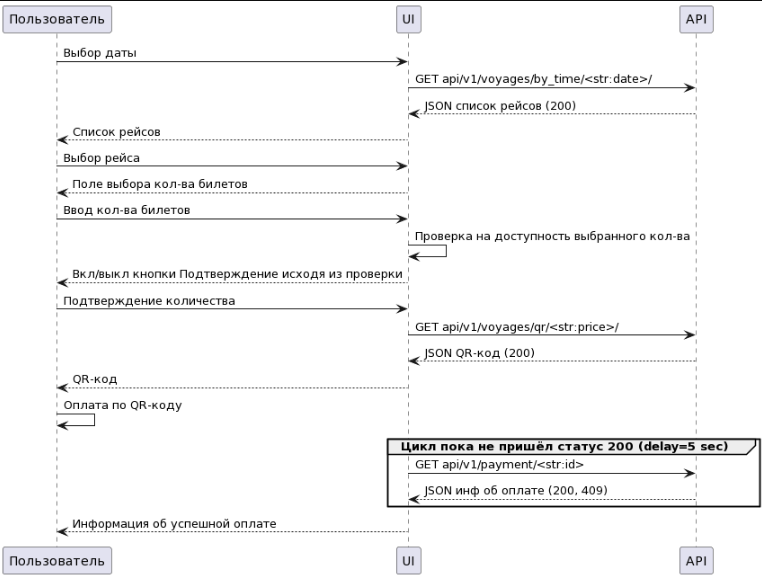


Рисунок 7 – Диаграмма последовательности для действия «Покупка билетов через интерактивный терминал»

Для назначения ТС и водителей на рейс была реализована диаграмма последовательности, которая представлена на рисунке 8.

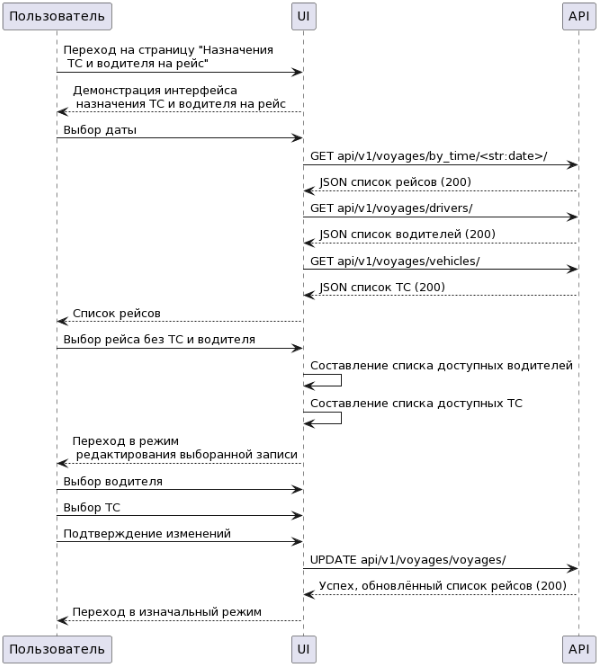


Рисунок 8 – Диаграмма последовательности для назначения водителей и ТС на рейс

Была реализована диаграмма состояний для сущности «Рейс», которая представлена на рисунке 9.

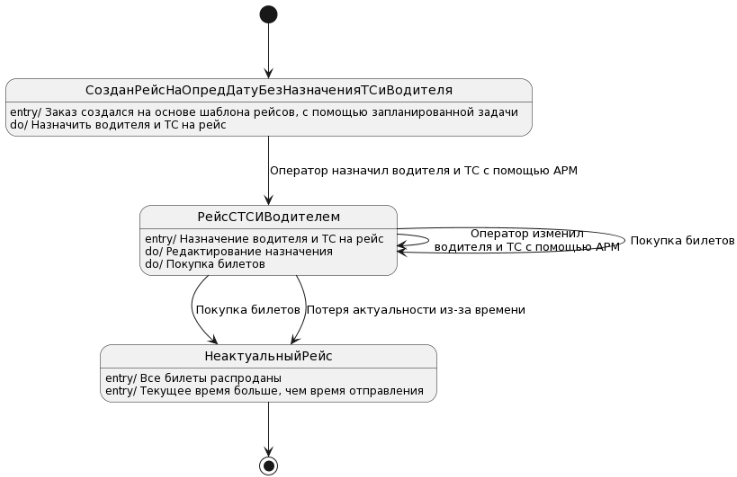


Рисунок 9 - Диаграмма состояний для сущности «Рейс»

# 4.2 Функциональная структура изделия. Алгоритмизация решения задачи автоматизации процессов предметной области

Была реализована цепочка «требования – функции – подсистема ПО –Компонент ПО» для требования ТЗ «Покупка билетов», которая представлена на рисунке 10. [12, 6]

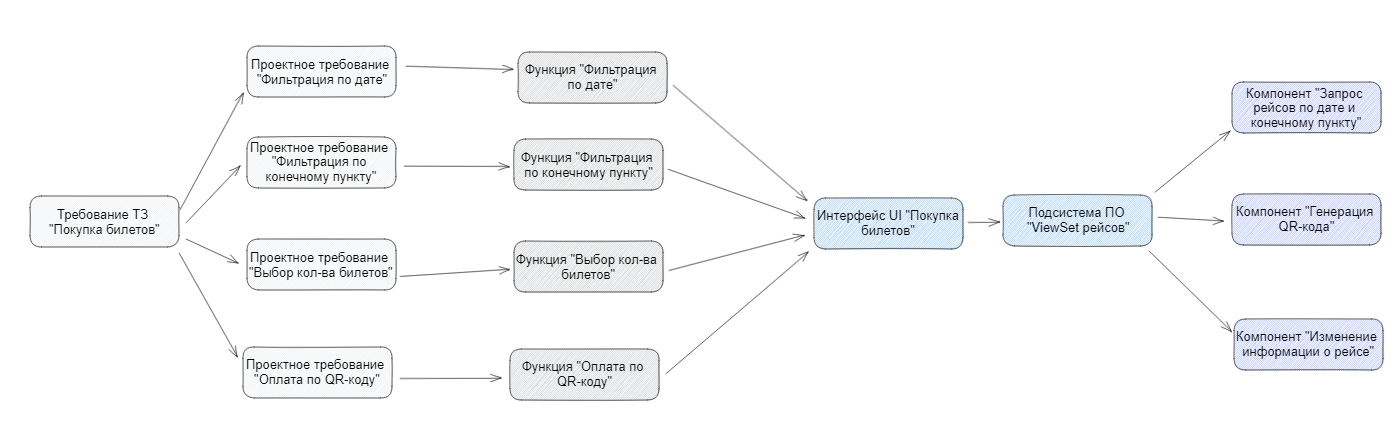


Рисунок 10 - Цепочка «требования – функции – подсистема ПО – Компонент ПО» для требования ТЗ «Покупка билетов»

Была реализована цепочка «требования – функции – подсистема ПО –Компонент ПО» для требования ТЗ «Демонстрация актуальных рейсов», которая представлена на рисунке 11. [12, 6]

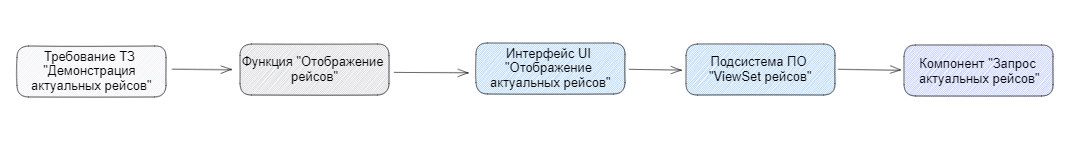


Рисунок 11 - Цепочка «требования – функции – подсистема ПО – Компонент ПО» для требования ТЗ «Демонстрация актуальных рейсов»

Была реализована цепочка «требования – функции – подсистема ПО –Компонент ПО» для требования ТЗ «АРМ оператора», которая представлена на рисунке 12. [12, 6]

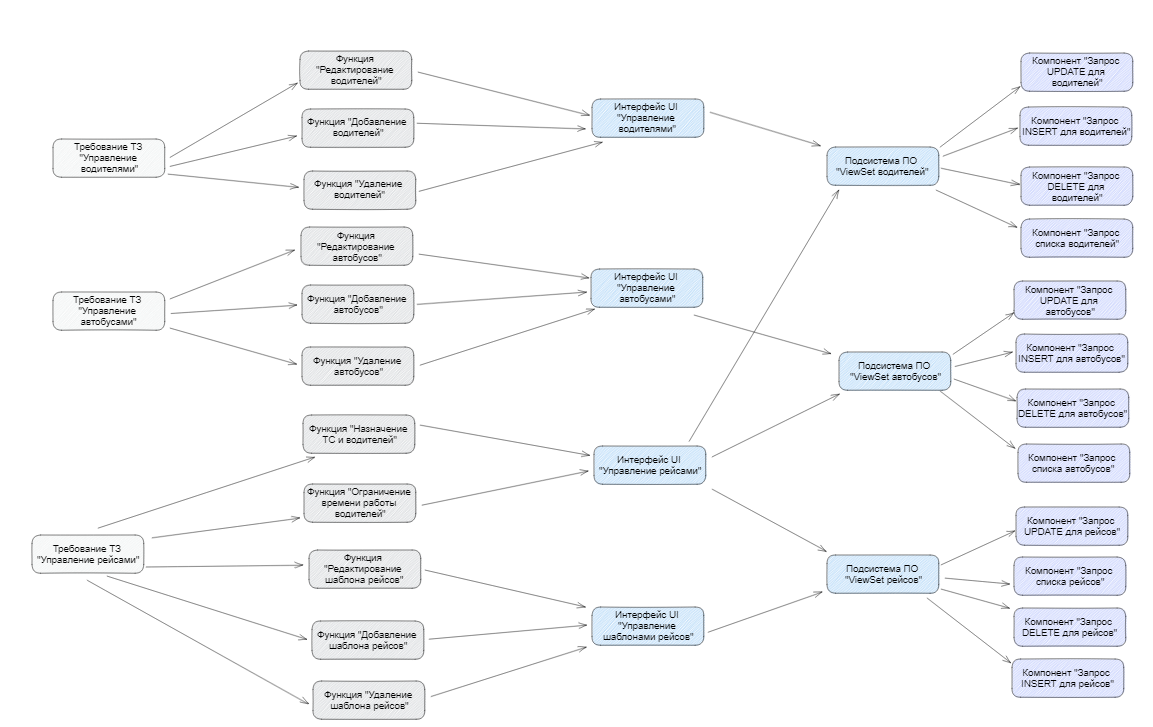


Рисунок 12 - Цепочка «требования – функции – подсистема ПО – Компонент ПО» для требования ТЗ «АРМ оператора»

Исходя из реализованных цепочек все требования ТЗ покрываются компонентами ПО.

# 4.3 Моделирование программной структуры и архитектурное описание изделия

Разрабатываемую систему можно разделить на две части: backend и frontend. Backend часть будет реализована на языке Python с использованием фреймворка Django. Frontend часть будет реализована на языке TypeScript с использованием фреймворка Angular. [8]

Двумя главными частями backend будут являться модели, виды, сериализаторы и пути. Модели необходимы для описания сущностей, с которым будет работать система. В видах описываются функции-запросы, которые будут исполняться при обращении к API. Сериализаторы необходимы для реализации запросов к базе данных. [9] В путях описываются URL адреса разрабатываемого API.

На frontend части необходимо реализовать CrudService для обращения к разрабатываемому API. Также, необходимы компоненты приложения, которые будут отвечать за отрисовку интерфейса.

Архитектура ПО представлена на рисунке 13.

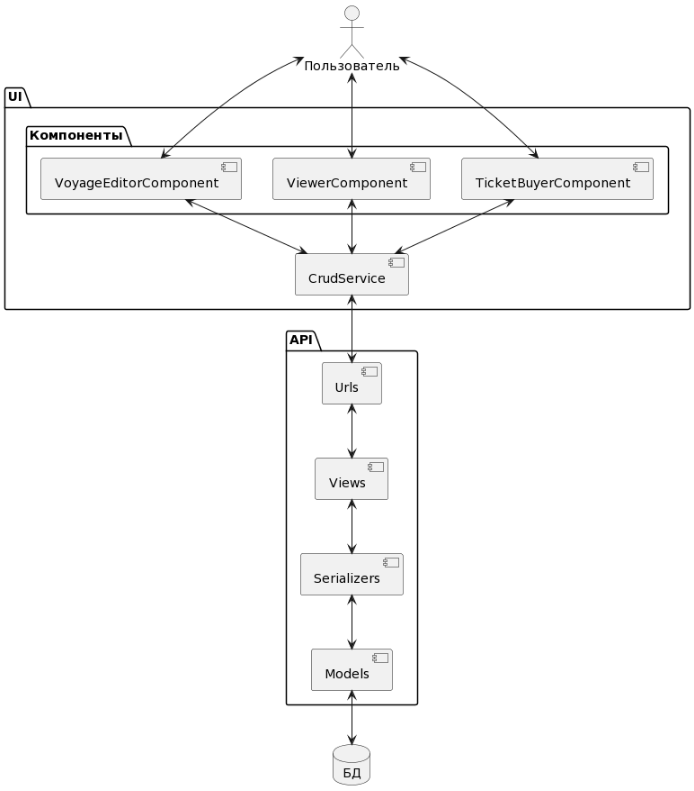


Рисунок 13 – Архитектура ПО

Техническая архитектура представлена на рисунке 14.

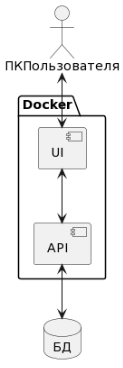


Рисунок 14 – Техническая архитектура

# 4.4 Сравнительный анализ аналогов изделия

Все аналоги разрабатывались под общие бизнес-процессы, которые невозможно назвать ультимативными. Почти все аналоги распространяются по подписочной системе за 1 ТС в автопарке, и не имеют всех необходимых функций в пакете.

Прямой аналог «Е-Автовокзал»:

|  |  |
| --- | --- |
| **Модуль** | **Цена, руб/месяц** |
| АРМ «Диспетчер» | 0 (входит в базовый функционал) |
| Программа «Электронное табло» | 1000 |
| Комплекс «АВ-терминал» для терминалов самообслуживания | 25 000 (за одну лицензию) |

Все решения не имеют открытого и гибкого API.

# 4.5 Перспективы применения технологических и технических решений

После реализации ПО останется открытое API, к которому можно будет свободно подключаться и использовать как backend для других потребностей, например: снятие отчётов, мобильная версия интерактивного терминала, мобильная версия табло.

Frontend не является таким же гибким инструмент, он покрывает исключительно требования ТЗ.

# 5 Описание организации работ с применением разрабатываемого изделия

# 5.1 Организация коллективной разработки

# 5.1.1 Ролевая модель команды

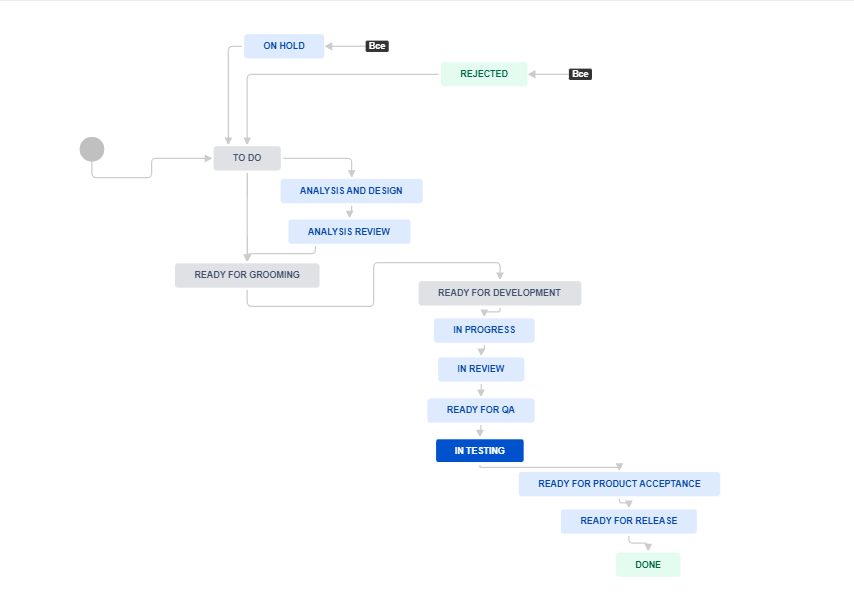
|  |  |
| --- | --- |
| **Роль** | **Обязанности** |
| Менеджер проекта | Руководство командой, обеспечения связи между подразделениями |
| Проектировщик | Анализ требований, разработка архитектуры, контроль качества |
| Разработчик (middle) | Контроль архитектурных и технических спецификаций, подбор инструментов и стандартов, написание программного кода, развертывание написанного кода в среде |
| DevOps-инженер | Контроль настроек среды, установка и настройка вспомогательных необходимых инструментов, развертывание написанного кода в среде |
| Тестировщик (middle) | Составление сценариев тестирования, выполнение сценариев тестирования, автоматизация тестирования |
| Аналитик | Анализ требований, контроль качества, разработка плана документирования, участие в тестировании, составление отчётности |
| Support-инженер | Составление нормативных документов поддержки решения, обработка поступающих заявок |

# 5.1.2 Правила выполнения разработки

Соглашения проектирования и реализации проекта: [5,8, 15]

* технические задания, реализованные аналитиками, должны быть одобрены и согласованы разработчиками и тестировщиками проекта;
* на основании технического задания тестировщиками должны быть составлены сценарии тестирования разрабатываемого модуля ПО;
* перед реализацией ПО разработчикам необходимо дать оценку сложности, от 1 до 12, где 1 – очень простая задача, а 12 – очень сложна задача, и запланированного времени реализации модуля или части функционала;
* каждый реализованный модуль или каждая функция должна быть покрыта Unit тестами;
* ведение реализации разрабатываемого ПО происходит на платформе Jira;
* ежедневные 15-ти минутные стендапы, на которых обсуждается текущий статус выполняемых задач;
* каждый первый и пятнадцатый день месяца происходит планирование спринта разработки;

Была предложена и согласована статусная модель разработки функций приложения.



* On hold – статус задачи, которая должна быть рассмотрена и одобрена тим-лидом;
* Rejected – задача, которая оказалась отвергнутой на любом этапе разработки;
* To-do – статус, указывающий на необходимость начала реализации разработки или проектирования;
* Analysis and design – статус аналитики и проектирования разрабатываемого функционала;
* Analysis review – ревью работы, выполненной аналитиком;
* Ready for grooming – готовность к оценке реализации разрабатываемого функционала;
* Ready for development – готовность к разработке функционала;
* In progress – разработка функционала;
* In review ­– ревью кода разработанного функционала;
* Ready for QA – готовность к тестированию;
* In testing – процесс тестирования;
* Ready for product acceptance – функционал успешно прошёл тестирование;
* Ready for release – функционал готов к публикации на сервер;
* Done – функционал успешно реализован и внедрён в ПО.

# 5.2 Развертывание и поставка программного обеспечения

Разрабатываемое решение будет развернуто посредством Docker на сервере заказчика. Непосредственно связь между машинами (сервер-терминал, сервер-ПК оператор) должна быть обеспечена инженерами заказчика. Внутри Docker контейнера также будет запущен супервизор pm2, который необходим для контроля технической стороны приложения, таких как состояние, логи, мониторинг ресурсов сервера. Развертывание базы данных происходит с помощью внутренних миграций ORM, который встроен в фреймворк Django.

# Приложение А. Перечень работ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ этапа** | **Стадия** | **Дата начала** | **Дата окончания** | **Результат** |
| 1 | сбор требований | 23.02.2023 | 01.03.2023 | готовое ТЗ, одобренное экспертом со стороны заказчика |
| 2 | разработка архитектуры приложения и БД | 02.03.2023 | 10.03.2023 | готовое техническое описание UML: диаграммы последовательности, диаграммы классов, диаграммы активности |
| 3 | написание кода приложения | 11.03.2023 | 25.03.2023 | готовый код приложения с покрытием тестами на 80% |
| 4 | испытание и тестирование системы | 26.03.2023 | 10.04.2023 | заключение о пройденном тестировании; демонстрация заказчику прототипа |
| 5 | написание технической документации | 11.04.2023 | 15.04.2023 | написанная техническая документация |
| 6 | эксплуатационное тестирование | 15.04.2023 | 30.04.2023 | успешно пройденное эксплуатационное тестирование; сформированный отчёт об успешности эксплуатационного тестирование; одобрение заказчика на запуск системы |
| 7 | передача и запуск системы | 30.04.2023 | 04.05.2023 | размещение системы на сервере заказчика; обученный персонал; передача всей технической документации |
| 8 | поддержка и работы по актуализации системы | 05.05.2023 | - | поддержка стабильной работы приложения |

# Приложение Б. Формализованные функциональные требования

|  |  |
| --- | --- |
| Формулировка требования №1 | |
| Краткое описание: | Покупка билета на рейс |
| Приоритет: | особо важный |
| Свойственные атрибуты качества: | Надёжность; Производительность |
| Схема проверки: | - успешная генерация штрихкода-билета. |
| Связь с элементом интерфейса: | Выпадающий календарь; Список рейсов |
| Роли, которым доступен элемент: | Администратор; Покупатель |

|  |  |
| --- | --- |
| Формулировка требования №2 | |
| Краткое описание: | Фильтрация списка билетов по дате |
| Приоритет: | высокий |
| Свойственные атрибуты качества: | Надёжность; Производительность |
| Схема проверки: | - в списке рейсов только рейсы, которые принадлежат только выбранной дате. |
| Связь с элементом интерфейса: | Выпадающий календарь; Список рейсов |
| Роли, которым доступен элемент: | Администратор; Покупатель |

|  |  |
| --- | --- |
| Формулировка требования №3 | |
| Краткое описание: | Фильтрация списка билетов по конечному пункту |
| Приоритет: | высокий |
| Свойственные атрибуты качества: | Надёжность; Производительность |
| Схема проверки: | - в списке рейсов только рейсы, которые принадлежат только выбранном конечному пункту. |
| Связь с элементом интерфейса: | Выпадающий список; Список рейсов |
| Роли, которым доступен элемент: | Администратор; Покупатель |

|  |  |
| --- | --- |
| Формулировка требования №4 | |
| Краткое описание: | Выбор количества билетов на выбранный рейс |
| Приоритет: | высокий |
| Свойственные атрибуты качества: | Надёжность; Производительность |
| Схема проверки: | - покупатель может выбрать от одного до максимального количества мест автобуса. |
| Связь с элементом интерфейса: | Счётчик; Кнопки ‘+’ и ‘-’ |
| Роли, которым доступен элемент: | Администратор; Покупатель |

|  |  |
| --- | --- |
| Формулировка требования №5 | |
| Краткое описание: | Оплата по QR-коду |
| Приоритет: | высокий |
| Свойственные атрибуты качества: | Надёжность; Производительность |
| Схема проверки: | - покупатель может выбрать от одного до максимального количества мест автобуса. |
| Связь с элементом интерфейса: | Счётчик; Кнопки ‘+’ и ‘-’ |
| Роли, которым доступен элемент: | Администратор; Покупатель |

|  |  |
| --- | --- |
| Формулировка требования №6 | |
| Краткое описание: | Демонстрация актуальных рейсов на табло |
| Приоритет: | особо важный |
| Свойственные атрибуты качества: | Надёжность |
| Схема проверки: | - демонстрация только актуальных будущих и текущих рейсов. |
| Связь с элементом интерфейса: | Список с рейсами |
| Роли, которым доступен элемент: | Администратор; Наблюдатель |

|  |  |
| --- | --- |
| Формулировка требования №7 | |
| Краткое описание: | Назначение ТС и водителя на рейс |
| Приоритет: | особо важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное обновление данных о водителе и ТС в рейсе. |
| Связь с элементом интерфейса: | Форма Drag&Drop |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №8 | |
| Краткое описание: | Фильтрация рейсов по дате |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - демонстрация только актуальных будущих рейсов. |
| Связь с элементом интерфейса: | Выпадающий календарь; Форма Drag&Drop |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №9 | |
| Краткое описание: | Генерация билетов |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - генерируется количество билетов строго равное количеству мест в автобусе. |
| Связь с элементом интерфейса: | Форма Drag&Drop |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №10 | |
| Краткое описание: | Отмена рейса |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное обновление данных о рейсе;  - пропадает возможность купить билеты на этот рейс. |
| Связь с элементом интерфейса: | Форма Drag&Drop |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №11 | |
| Краткое описание: | Редактирование списка ТС |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное обновление данных о ТС; |
| Связь с элементом интерфейса: | Форма изменения ТС |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №12 | |
| Краткое описание: | Удаление ТС |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное удаление данных о ТС; |
| Связь с элементом интерфейса: | Форма изменения ТС |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №13 | |
| Краткое описание: | Добавление ТС |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное добавление данных о новом ТС; |
| Связь с элементом интерфейса: | Форма изменения ТС |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №14 | |
| Краткое описание: | Изменение имеющихся ТС |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное изменение данных о ТС; |
| Связь с элементом интерфейса: | Форма изменения ТС |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №15 | |
| Краткое описание: | Редактирование списка водителей |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное обновление данных о водителях; |
| Связь с элементом интерфейса: | Форма изменения водителей |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №16 | |
| Краткое описание: | Удаление водителей |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное удаление данных о водителе; |
| Связь с элементом интерфейса: | Форма изменения водителей |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №17 | |
| Краткое описание: | Добавление водителя |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное добавление данных о новом водителе; |
| Связь с элементом интерфейса: | Форма изменения водителей |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №18 | |
| Краткое описание: | Изменение имеющихся водителей |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное изменение данных о водителей; |
| Связь с элементом интерфейса: | Форма изменения водителей |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №19 | |
| Краткое описание: | Редактирование списка операторов |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное обновление данных о операторах; |
| Связь с элементом интерфейса: | Форма изменения операторов |
| Роли, которым доступен элемент: | Администратор |

|  |  |
| --- | --- |
| Формулировка требования №20 | |
| Краткое описание: | Удаление оператора |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное удаление данных о операторе; |
| Связь с элементом интерфейса: | Форма изменения операторов |
| Роли, которым доступен элемент: | Администратор |

|  |  |
| --- | --- |
| Формулировка требования №21 | |
| Краткое описание: | Изменение имеющегося оператора |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное изменение данных о операторе; |
| Связь с элементом интерфейса: | Форма изменения операторов |
| Роли, которым доступен элемент: | Администратор |

|  |  |
| --- | --- |
| Формулировка требования №22 | |
| Краткое описание: | Добавление оператора |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное добавление нового оператора; |
| Связь с элементом интерфейса: | Форма изменения операторов |
| Роли, которым доступен элемент: | Администратор |

|  |  |
| --- | --- |
| Формулировка требования №23 | |
| Краткое описание: | Редактирование списка шаблонов рейсов |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное обновление данных о шаблонах рейсов; |
| Связь с элементом интерфейса: | Форма изменения шаблонов рейсов |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №24 | |
| Краткое описание: | Удаление шаблона рейсов |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное удаление шаблона рейсов; |
| Связь с элементом интерфейса: | Форма изменения шаблонов рейсов |
| Роли, которым доступен элемент: | Администратор; Оператор |

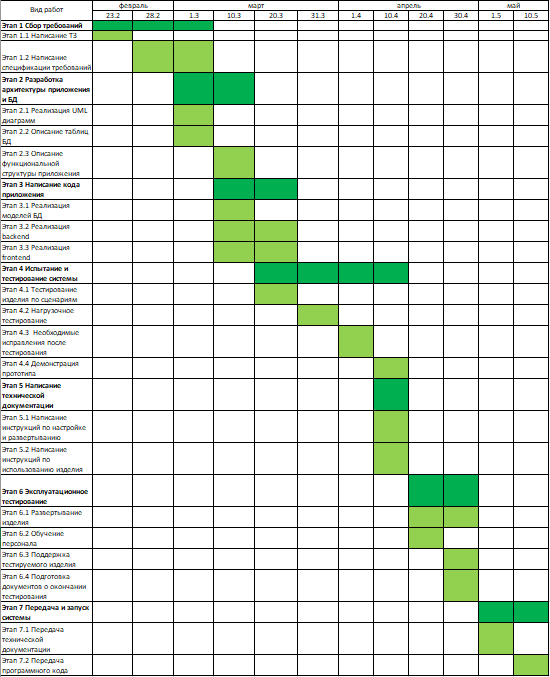
|  |  |
| --- | --- |
| Формулировка требования №25 | |
| Краткое описание: | Изменение имеющегося шаблона рейсов |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное изменение шаблона рейсов; |
| Связь с элементом интерфейса: | Форма изменения шаблонов рейсов |
| Роли, которым доступен элемент: | Администратор; Оператор |

|  |  |
| --- | --- |
| Формулировка требования №26 | |
| Краткое описание: | Добавление шаблона рейсов |
| Приоритет: | важный |
| Свойственные атрибуты качества: | Надёжность; Производительность; Доступность |
| Схема проверки: | - успешное добавление шаблона рейсов; |
| Связь с элементом интерфейса: | Форма изменения шаблонов рейсов |
| Роли, которым доступен элемент: | Администратор; Оператор |

# Приложение В. Структура данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Элементы данных** | **Название элемента** | **Описание** | **Структура или тип данных** | **Значение** |
| Номер рейса | VoyageNumber | Номер рейса, по которому будут ориентироваться пассажиры | Строка | Появляется при первой миграции, как данные инициализации. Получаются от заказчика |
| Время отправления рейса | TimeDeparture | Время, во сколько рейс должен отправляться | Время | Появляется при первой миграции, как данные инициализации. Получаются от заказчика |
| Время окончания рейса | EndTimeDeparture | Время, во сколько рейс должен закончится, а автобус должен вернуться | Время | Появляется при первой миграции, как данные инициализации. Получаются от заказчика |
| Сколько дней занимает рейс | Days | Количество дней, сколько длится рейс | Целое число | Появляется при первой миграции, как данные инициализации. Получаются от заказчика |
| Актуальность рейса | IsActive | Актуален ли рейс. Есть возможность отключить и не генерировать новые рейсы данного типа | Логический | Появляется при первой миграции, как данные инициализации. Получаются от заказчика |
| ID автобуса | BusId | Идентификационный номер автобуса | Целое число | Генерируется автоматически БД |
| Транспортный номер автобуса | LicensePlate | Транспортный номер, под которым ездит автобус | Строка | Заводится оператором или администратором. Имеет строгую маску: 1 буква, 3 цифры, 2 буквы, 2-3 цифры. Например: А123АА66 |
| Количество мест | SitPlaces | Количество мест в автобусе | Целое число | Заводится оператором или администратором |
| Находится ли автобус в ремонте | Broken | Состояние автобуса | Логический | Заводится оператором или администратором |
| ID водителя | DriverId | Идентификационный номер водителя | Целое число | Генерируется автоматически БД |
| Имя водителя | FirstName | Имя водителя | Строка | Заводится оператором или администратором |
| Фамилия водителя | SecondName | Фамилия водителя | Строка | Заводится оператором или администратором |
| Отчество водителя | ThirdName | Отчество водителя | Строка | Заводится оператором или администратором |
| Болен ли водитель | Illness | Болен ли водитель, доступен ли для назначения | Логический | Заводится оператором или администратором |
| Количество часов, отработанных водителем за неделю | HoursWorked | Часы, сколько отработал водитель | Целое число | Рассчитывается исходя из назначений на рейсы. Не может превышать 40 часов. |
| Дата отправления рейса | DateDeparture | Дата, когда рейс должен отправится | Дата | Генерируется каждый день на Т+2 |
| Количество оставшихся билетов | AvailableTickets | Количество билетов, которые могу приобрести пассажиры | Целое число | Генерируется после назначенного ТС на рейс. Уменьшается после приобретения билета на рейс |
| Имя оператора | FirstName | Имя оператора | Строка | Задаётся администратором |
| Фамилия оператора | SecondName | Фамилия оператора | Строка | Задаётся администратором |
| Отчество оператора | ThirdName | Отчество оператора | Строка | Задаётся администратором |
| ID рейса | IdVoyage | Идентификационный номер рейса | Целое число | Генерируется БД |

# Приложение Г. Уточнение сетевого плана графика



# Перечень литературы и документов, используемых при разработке пояснительной записке к техническому проекту

1. Техническое задание на реализацию разработки программного обеспечения для автоматизации/информационной системы автобусного вокзала.
2. Спецификация требований на реализацию разработки программного обеспечения для автоматизации/информационной системы автобусного вокзала.
3. IEEE Swebok V 3.0 Guide to the Software Engineering. Body of Knowledge. A Project of the IEEE Computer Society. Inetrnet resource. URL: www.computer.org (дата обращения: 10.03.2022)
4. Вигерс К., Битти Д. Разработка требований к программному обеспечению/Пер. с англ. – М.: Издательство "БХВ-Петербург", 2014. – 736 с.
5. Карпенко С.Н. Введение в программную инженерию. Учебно-методические материалы по программе повышения квалификации «Информационные технологии и компьютерное моделирование в прикладной математике». Нижний Новгород, 2007, 103 с.
6. ГОСТ 2.103-68 Единая система конструкторской документации. Стадии разработки.
7. ГОСТ 2.120-73 Единая система конструкторской документации. Технический проект.
8. Маглинец Ю.А. Разработка информационных систем. Часть 1, Структурные методы. – Красноярск.: Кларитеанум, 2004. – 120 с.
9. Белов, В.В. Проектирование информационных систем: Учебник / В.В. Белов. - М.: Академия, 2018. - 144 c.
10. Гвоздева, Т.В. Проектирование информационных систем. Стандартизация: Учебное пособие / Т.В. Гвоздева, Б.А. Баллод. - СПб.: Лань, 2019. - 252 c.
11. Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: Методы и средства проектирования информационных систем и технологий. Инструментальные средства информационных систем: Учебное пособие / С.А. Мартишин, В.Л. Симонов,. - М.: ИД ФОРУМ, НИЦ Инфра-М, 2012. - 160 c.
12. Федоров, Н.В. Проектирование информационных систем на основе современных CASE-технологий / Н.В. Федоров. - М.: МГИУ, 2008. - 280 c.
13. Мюллер, Р.Д. Проектирование баз данных и UML / Р.Д. Мюллер. - М.: Лори, 2013. - 420 c.
14. Эмблер, С. Рефакторинг баз данных: эволюционное проектирование / С. Эмблер, П. Садаладж. - М.: Вильямс, 2007. - 672 c.
15. В.П. Романов, Н.З. Емельянова, Т.Л. Партыка Проектирование экономических информационных систем. Методологии и современные технологии. - М: Экзамен, 2005.- 256 с.

# Вывод

Была реализована пояснительная записка к техническому проекту. Область применения и концепт разрабатываемого изделия был комплексно рассмотрен с нескольких сторон. Была описана актуальность разрабатываемого изделия, обозначены проблемы, которые присутствуют в текущей схеме работы предприятия, а также решения, которые будут имплементированы в изделие для их устранения. Была формализована цель технического проектирования, которая звучит как повышение эффективности работы предприятия, значительное улучшение опыта получения услуг для пассажиров, упрощение и гарантирование верного учёта ресурсов (водители, автобусы). Был проведен системный анализ, выявлены интересанты, описана среда.

Требования к изделию были форматизированы и комплексно изучены на нестыковки и упущения, а также возможность их реализации. При помощи UML диаграмм и функциональной цепочки была описана структура и архитектура разрабатываемого изделия.

Была описана организация коллективной разработки, с точки зрения ответственности и обязанностей, которые буду возлагаться при разработке.