

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

А. Э. Зянчурин

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА
СПЕЦИФИКАЦИЙ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

по курсу: МЕТОДОЛОГИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4232М

подпись, дата

В. Ф. Губайдулин

инициалы, фамилия

Санкт-Петербург 2023

Цель работы:

Получение практических навыков, необходимых при обследовании объекта автоматизации и разработке спецификаций требований к изделию (автоматизированной системе или программному обеспечению).

Вариант 5:

Разработка программного обеспечения для автоматизации/информационной системы автобусного вокзала.

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

Спецификация требований

Спецификация требований

1. Общее описание изделия

1.1 Назначение, соглашения и границы проекта

1.1.1 Назначение

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

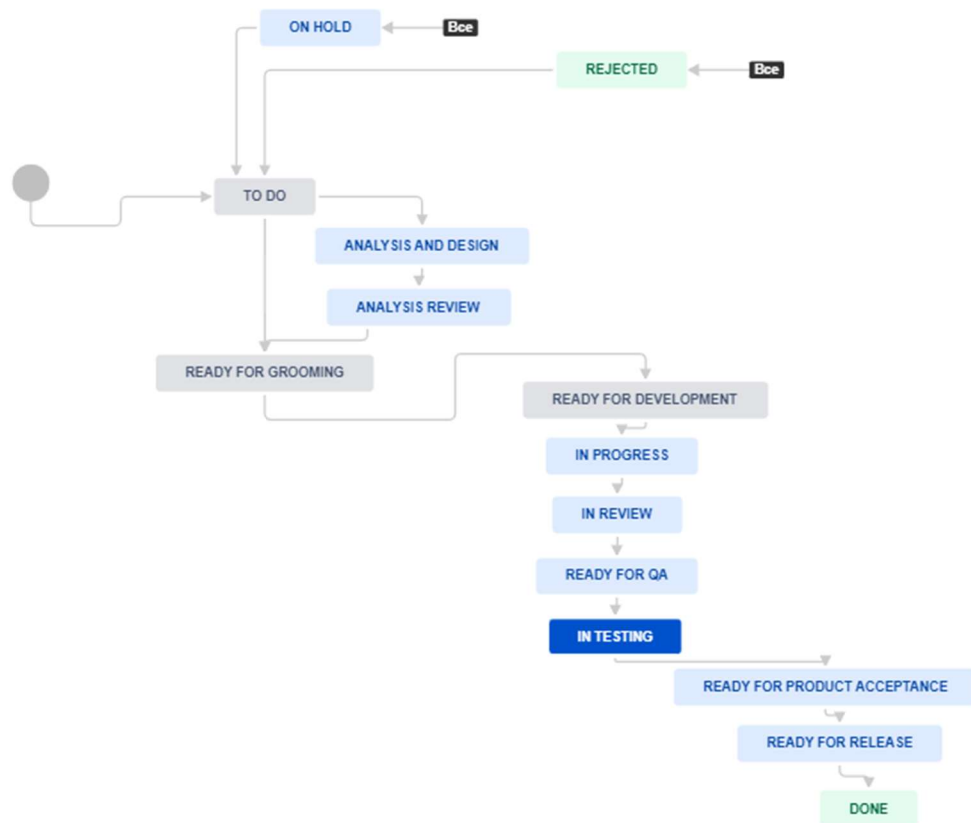
- покупка билетов;
- демонстрация актуальных рейсов;
- назначение водителей и ТС на рейсы.

1.1.2 Соглашения

Соглашения проектирования и реализации проекта:

- 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 сложная 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.

1.1.3 Границы проекта

Границы проекта чётко обозначены выше в пункте «Назначение».

Остальные процессы автобусного вокзала не включены в реализацию системы автоматизации.

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

Покупка билетов будет производиться 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 производиться 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.

1.2 Определение вариантов использования изделия

Для выполнения анализа информации о предметной области была составлена матрица CRUD, представленная в таблице 1.

Таблица 1 – Матрица CRUD

Сущность	Рейс	Шаблон рейса	Автобус	Водитель	Оператор
Варианты исп-ия					

Покупка билета	U				
Запрос от табло	R		R	R	
Назначение ТС и водителя на рейс	U		R	R	
Редактирование ЛК оператора					C, U, D
Редактирование шаблона рейса		C, U, D			
Редактирование списка автобусов			C, U, D		
Редактирование списка водителя				C, U, D	
Смена дня (наступление 00:00)	C	R			
Отмена рейса	U		U	U	

При составлении матрицы CRUD не было обнаружено недостающих требований.

1.3 Дерево функций изделия

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 1.

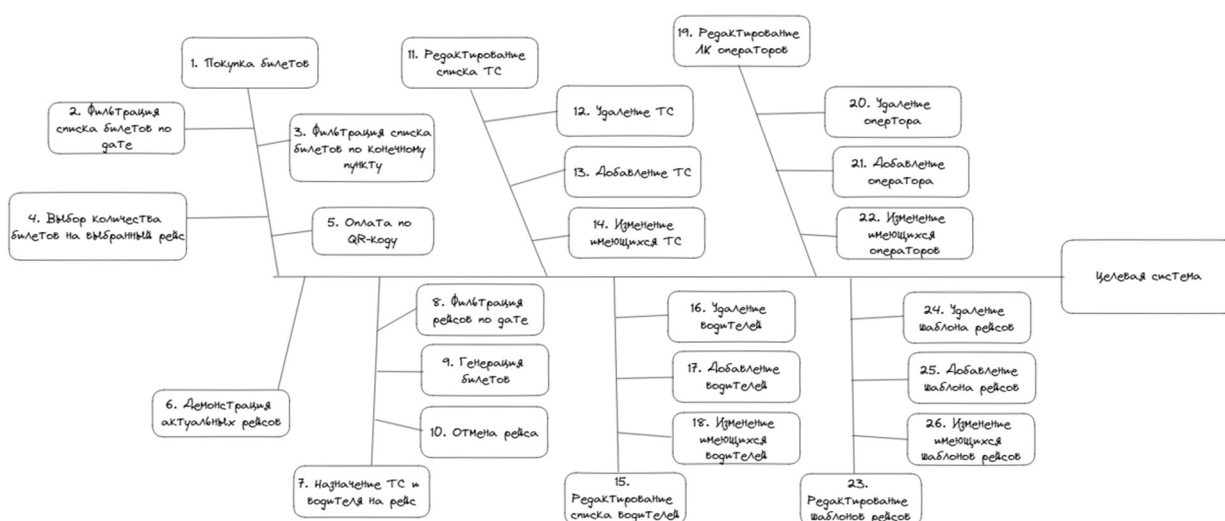


Рисунок 1 – Дерево функций изделия

1.4 Операционная среда

ОС сервера для запуска приложения: Debian 11. Инструмент развёртывания и управления: Docker (или pm2).

ОС сервера БД: Debian 11. Движок БД: MySQL 8.0.

ОС интерактивных терминалов по покупке билетов: любой из Windows 11, Android 9.0.

ОС информационного табло: любой из Windows 11, Android 9.0.

ОС АРМ оператора: Windows 11, Debian 11.

Поддерживаемые браузеры: Edge (версии 98-112), Firefox (версии 100-111), Opera (версии 83-95), Chrome (версии 97-111).

2 Функциональные требования

Для формализации функциональных требований были разработаны карточки с описанием их свойств в виде таблиц.

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 выбранному 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 водителях;
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

3. Требования к данным

3.1 Словарь данных предметной области

Был сформирован словарь данных для идентификации объектов в исследуемой предметной области.

3.1.1 Простейшие элементы данных

- Номер рейса;
- Время отправления рейса;
- Время окончания рейса;
- Сколько дней занимает рейс;
- Актуальность рейса;
- ID автобуса;
- Транспортный номер автобуса;
- Количество мест;
- Находится ли автобус в ремонте;
- ID водителя;
- Имя водителя;
- Фамилия водителя;
- Отчество водителя;
- Болен ли водитель;
- Количество часов, отработанных водителем за неделю;
- Дата отправления рейса;
- Количество оставшихся билетов;
- Имя оператора;
- Фамилия оператора;
- Отчество оператора;
- ID рейса.

3.1.2 Структура данных

Была составлена таблица со структурой данных, представленная в таблице 2.

Таблица 2 – 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 закончиться, 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: A123AA66
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 отправиться	Дата	Генерируется каждый день на T+2
Количество оставшихся билетов	AvailableTickets	Количество билетов, которые могут 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

3.2 Логическая модель

Была реализована UML диаграмма классов, представлена на рисунке 2.

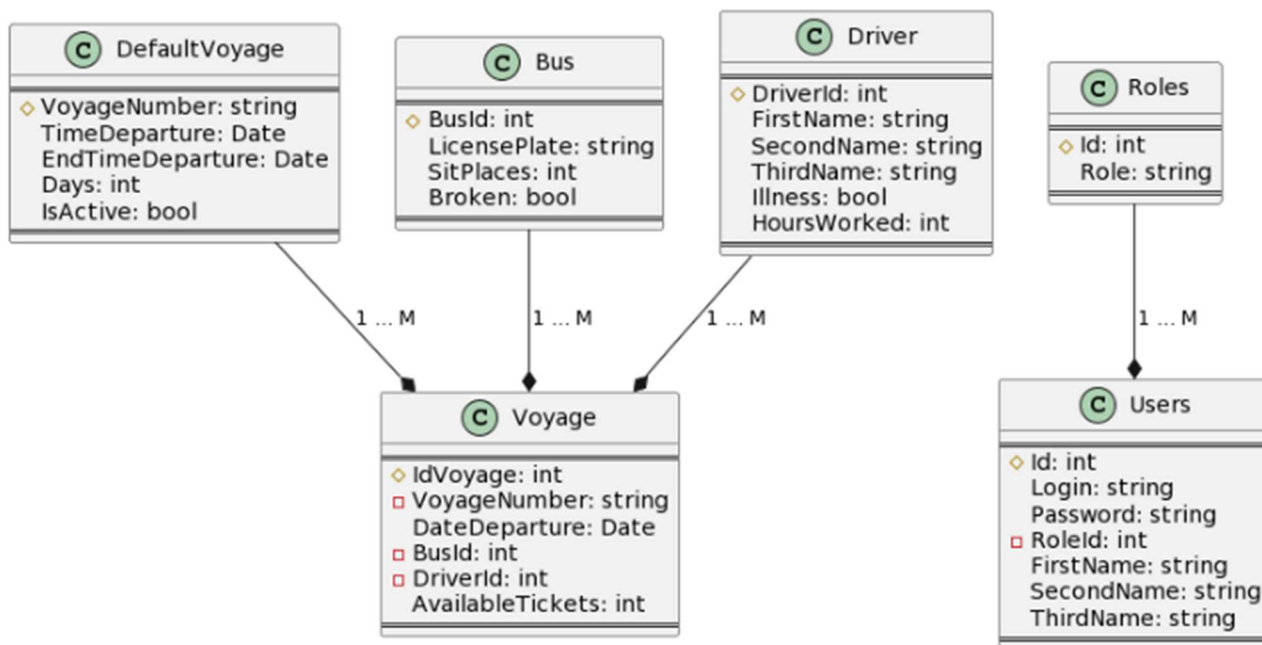


Рисунок 2 – UML-диаграмма классов

3.3 Журналирование событий

Были разработаны 5 форматов журналирования для ПО и его подсистем:

- Добавление таблицы TransactionHistory. В таблице будут храниться отметки всех ключевых транзакций в БД (покупка билета, назначение водителя и ТС на рейс, создание рейса).
- Запись журнала событий API. Формирование файла, в котором будет писаться журнал событий по успешности завершения сервисов внутри.
- Запись логов Front. Формирование файла, в котором будет писаться журнал событий front-end части приложения. Записываются ошибки, которые возникли при отработке кода, отчётность о запросах в API (код,

тело ответа).

- Запись журнала событий запланированных задач, реализуемых в API.
- Добавление таблицы Monitoring. При возникновении ошибки выполнения запроса, записывается строка с описанием ошибки и временем её возникновения.

3.4 Сбор, целостность, хранение и утилизация данных

3.4.1 Сбор

Обращение к таблице ограничено 10 секундами. Если запрос не успел выполниться за отведённое время, он считается ошибочным и будет исправлен инженерами производительности на стороне заказчика.

3.4.2 Целостность

Каждые сутки в 00:00 генерируются рейсы из шаблона рейсов на T+5.

Если начинается новая неделя, также сбрасываются рабочие часы водителя, если они больше 20.

3.4.3 Хранение

Каждый день в 22:00 запускается резервное копирование всей БД. Резервный файл копируется на другой сервер, предназначенный только для хранения резервных файлов. На основном сервере файл может храниться 3 месяца, после этого он должен быть удалён. На сервере, предназначенном для резервных файлов, файл может храниться 6 месяцев, после этого времени также должен быть удалён.

Все журналы событий должны храниться в системе максимум 7 дней.

3.4.4 Утилизация

Таблица Voyage должна быть очищена до 1 числа текущего месяца, раз в 3 месяца 15-го числа.

4. Требования к интерфейсам

4.1 Требования к пользовательским интерфейсам

4.1.1 Общие требования к пользовательским интерфейсам

Разрабатываемое ПО должно удовлетворять характеристикам, соответствующим ГОСТ Р ИСО 9241-151-2014, а также рекомендательный характер несут стандарты: «Требования к дисплеям при наличии отражений» – ИСО 9241-300; «Требования к неклавиатурным устройствам ввода» – ИСО 9241-400; «Представление информации» – ИСО 9241-111 и ИСО 9241-141; «Диалоги заполнения форм» – ИСО 9241-134.

Основным шрифтом разрабатываемого приложения будет Helvetica Regular 400. В качестве значков будут использоваться встроенные значки front-end библиотеки Angular из списка Mat-Icon. На каждом интерфейсе посередине наверху следует расположить название или логотип автовокзала. Исключить возможность повторного нажатия кнопок, инициализирующих запрос к API, до получения ответа от API (деактивация кнопок, спинер). Кнопки, подтверждающие действие пользователя, будут иметь зелёный фон и белый цвет шрифта. Кнопки, отменяющие действие пользователя, будут иметь красный фон и белый цвет шрифта. Кнопки, несущие промежуточные решения пользователя, будут иметь серый фон и черный цвет шрифта. Все всплывающие диалоговые окна будут затемнять основную заднюю часть веб-страницы. При неуспешном запросе к API необходимо выводить всплывающее сообщение о сбое.

Все интерфейсы должны быть адаптированы под соотношения сторон 16:9 и разрешения: 1280x720, 1366x768, 1920x1080.

Для каждого пользователя будет определён свой стандартный URL адрес для работы с веб-приложением, за границы которого он не сможет выйти, из-за этого навигационное меню между модулями приложения реализовывать категорически не стоит.

В разрабатываемом ПО должны отсутствовать любые сочетания клавиш.

В системе будут определены следующие поля для ввода информации:

- Календарь – запретить ввод с клавиатуры, ввод в поле путём выбора в календаре;
- Строковые – отсутствуют ограничения и валидация;

- Числовые – валидация исключительно на числовые значения в поле.

Специальные возможности для пользователей с проблемами со зрением, различением цвета и другими ограничениями не предусмотрены.

4.1.2 Требования к пользовательскому интерфейсу «Покупка билетов»

Пользователя должно встречать приветственное окно с кнопкой «Приобрести билет», после нажатия на которую пользователь видит список с актуальными рейсами, на которые ещё не закончились билеты. В самом рейсе содержится информация: конечный пункт, время отправления, время прибытия, количество доступных билетов и цена. Пользователь может пользоваться фильтрами по дате и по названию конечного пункта. Фильтра по дате представляется из себя поле типа «Календарь». Фильтр по названию конечного пункта состоит из выпадающего списка, который формируется на основании доступных рейсов. При нажатии на рейс пользователю поверх элемента рейса всплывает сообщение с просьбой выбрать количество билетов, с динамической ценой, зависимой от выбранного количества билетов, а также кнопка о подтверждении выбранного количества. После нажатия, пользователь видит QR-код для оплаты и кнопку отмены. При нажатии на кнопку отмены пользователь попадает на приветственное окно. При оплате по QR-коду пользователь получает чек об оплате и перенаправляется на приветственное окно.

4.1.3 Требования к пользовательскому интерфейсу «Информационное табло»

При открытии модуля «Информационное табло» пользователь видит список с актуальными будущими рейсами. Каждый рейс имеет информацию: конечный пункт, время отправления, время прибытия. Табло обновляется раз в 1 минуту для удаления устаревших данных с табло.

4.1.4 Требования к пользовательскому интерфейсу «АРМ оператора»

При открытии модуля «АРМ оператора» пользователя встречает навигационное меню: прикрепление водителя к ТС, прикрепление ТС и водителя к рейсу, редактирование шаблонов рейсов.

Интерфейс редактирование шаблонов рейсов демонстрирует список шаблонов рейсов. В информации о рейсе информация: время отправления, время прибытия, активность. В конце списка напротив каждого элемента присутствуют кнопки редактировать или удалить. При нажатии на кнопку редактирования появляется возможность редактировать элементы выбранного рейса. Вместо кнопок редактировать и удалить появляются кнопки подтверждения или отмены. При нажатии на кнопку удалить пользователю всплывает окно подтверждения удаления элемента.

Прикрепление водителя к ТС производится при помощи списка доступных водителей. В информации о водителе информация: имя, фамилия, отчество, состояние здоровья, отработанное время, ТС. В конце списка напротив каждого элемента присутствуют кнопки редактировать или удалить. При нажатии на кнопку редактирования появляется возможность редактировать элементы выбранного водителя, кроме отработанного времени. Вместо кнопок редактировать и удалить появляются кнопки подтверждения или отмены. При нажатии на кнопку удалить пользователю всплывает окно подтверждения удаления элемента.

Прикрепление водителя и ТС к рейсу производится с помощью технологии Drag&Drop, путём переноса ТС с водителем на рейс. При невозможности назначения ТС на рейс (состояние здоровья водителя, состояние ТС и др.) должна всплывать соответствующая ошибка.

4.2 Требования к информационному взаимодействию

Связь с базой данных будет осуществлена с помощью Django ORM. Подключаемыми библиотеками со стороны Python будут: Django, copy, json,

Django-scheduler, asyncio, aiohttp. Подключаемыми библиотеками со стороны TypeScript будут: Angular, ngx-spinner. С помощью Django будет реализован RESTfull API, к которому сможет обращаться front-end. Обмен данными будет организован с помощью JSON-файлов.

Так как back-end и front-end части будут находится исключительно в одной частной сети, авторизацию между частями не требуется.

4.3 Требования к аппаратным интерфейсам

RESTfull API будет использовать протокол обмена HTTP.

Входными данными для запросов GET или DELETE могут быть id или дата.

Входными данным для запросов PUT и POST могут быть разнообразные JSON файлы, которые на стороне Django будут реализованы как классы DTO.

Выходными данными будут являться статус возвращаемого сообщения (200, 404, 500), а также JSON файлы которые на стороне Django будут реализованы как классы DTO, а на стороне Angular как модели.

5 Нефункциональные требования и атрибуты качества

5.1 Производительность

Любая обработка ввода пользователем информации, кроме ввода информации, требующих запроса к API, должна составлять меньше или равно 1 секунды. Обработка ввода пользователем информации, которая требует запрос к API, не должна превышать 5 секунд. При успешности запроса, пользователь должен видеть сообщение об успешности совершенной операции. При неуспешности, должен видеть сообщение об ошибке.

5.2 Доступность

Сервисы должны быть доступны круглосуточно каждый день, за исключением запланированных технических работ.

Модуль покупки билетов должен быть открыт на 5 интерактивных терминалах одновременно.

Модуль табло должен быть открыт на 1 табло.

Модуль АРМ оператора должен быть открыт на 1 ПК оператора.

5.3 Целостность

При ошибке запроса и возникновении ошибки 500 в API администратор будет уведомлён с помощью email сообщения. Пользователь увидит ошибку и рекомендацию не пользоваться системой из-за технических неполадок. При ошибочном выполнении запланированной задачи, администратор также будет уведомлён с помощью email сообщения.

5.4 Совместимость

Поддерживаемые браузеры: Edge (версии 98-112), Firefox (версии 100-111), Opera (версии 83-95), Chrome (версии 97-111).

5.5 Надёжность

Система может повести себя непредсказуемо и работать с ошибками только из-за недостатка данных в базе данных. Если база данных имеет достаточную наполненность, система будет работать корректно круглосуточно.

5.6 Безопасность

Система имеет механизмы авторизации и аутентификации по ролям. Каждая роль имеет доступ исключительно к своему модулю разрабатываемого ПО. Все пароли шифруются с помощью SHA256. Исключить возможность SQL-инъекций.

5.7 Устойчивость

В неожиданных условиях работы системы присутствует механизм

уведомления администратора, а также механизм мониторинга, который проверяет наличие необходимых данных в базе данных. Мониторинг реализован на базе Zabbix.

5.8 Масштабируемость

Система без проблем справится с числом пользователей до 50 человек одновременно, сможет принимать порядка 100-200 транзакций в минуту.

5.9 Хранение данных

Все данные в базе данных должны храниться на всем промежутке времени жизни системы. Таблица Voyage должна быть очищена до 1 числа текущего месяца, раз в 3 месяца 15-го числа. На основном сервере для резервных копий файл может храниться 3 месяца, после этого он должен быть удалён. На сервере, предназначенном для резервных файлов, файл может храниться 6 месяцев, после этого времени также должен быть удалён.

Все журналы событий должны храниться в системе максимум 7 дней.

5.8 Удобство пользователей

Модуль покупки билетов должен быть адаптирован под сенсорные экраны.

Модуль табло должен быть адаптирован под размер экрана в 60-80 дюймов, вся информация должна быть читаема.

Модуль АРМ оператора должен иметь всплывающие подсказки при наведении на кликабельный элемент интерфейса.

5.9 Повторное использование компонентов

Все запросы должны быть разбиты на контроллеры, для каждого контроллера должен быть реализован Swagger. В дополнение к Swagger необходимо написать документация по использованию разрабатываемого API отдельно от front-end части ПО.

5.10 Расширяемость

Без проблем могут быть расширены реализуемые таблицы в базе данных, при условии, что основные атрибуты не будут подвергнуты изменениям.

5.11 Переносимость

Система способна стабильно работать только на платформах, которые были обозначены выше в требованиях.

5.12 Взаимодействие между компонентами

Все модули приложения работают через единый API по средством протокола HTTP. Front-end часть приложения работает с помощью HTTPS.

5.13 Поддержка системы

Быстрый анализ ошибок с помощью ведения журнала событий, простота при публикации проекта, постоянный мониторинг системы с помощью стабильной внешней системы Zabbix.

5.14 Тестирование

Разработчикам необходимо в обязательном порядке написать Unit тесты, которые будут покрывать минимум 80% написанного кода. Перед публикацией каждой новой версией приложения все, написанные ранее, Unit тесты должны быть успешно пройдены.

Тестировщикам следует составить сценарии тестирования всех частей и модулей приложения.

Инженерами по производительности следует составить и выполнить план нагрузочного тестирования, результаты оформить в виде отчёта.

5.15 Локализация

Система не требует локализации, так как нацелена исключительно на внутренний рынок.

5.16 Версионность

Каждая новая версия приложения должна быть помечена при публикации в Swagger.

5.17 Совместимость с предыдущими версиями

Поддерживать старые версии приложения нет необходимости.

6 Нормативные требования

6.1 Требования стандартов

Стандарт проектирования здания автовокзалов - СП 462.1325800.2019.

Приказ Министерства транспорта РФ от 2 октября 2020 г. № 406 “Об утверждении минимальных требований к оборудованию автовокзалов и автостанций”.

На основании приказа Министерства транспорта РФ от 16 октября 2020 г. № 424 “Об утверждении Особенности режима рабочего времени и времени отдыха, условий труда водителей автомобилей” система обеспечит точный подсчёт времени работы водителя, и исключит фактор переработок.

6.2 Требования защиты информации

Требования составлены на основании приказа ПРИКАЗ 11 февраля 2013 г. N 17 ОБ УТВЕРЖДЕНИИ ТРЕБОВАНИЙ О ЗАЩИТЕ ИНФОРМАЦИИ, НЕ СОСТАВЛЯЮЩЕЙ ГОСУДАРСТВЕННУЮ ТАЙНУ, СОДЕРЖАЩЕЙСЯ В ГОСУДАРСТВЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ.

Разрабатываемая система не хранит в себе никакую конфиденциальную информацию пользователей, водителей, операторов. В базе данных хранится

только вся соответствующая информация о рейсах.

Система не взаимодействует с системами и библиотеками не имеющими лицензию на свободное коммерческое использование, а также не имеющими разрешение на использование в РФ по политическим соображениям.

6.3 Требования к локализации

В системе не предусмотрена никакая локализация, однако в будущем есть возможность по реализации front-end части на английском языке.

Вывод:

Получены практические навыки, необходимые при обследовании объекта автоматизации и разработке спецификаций требований к изделию (автоматизированной системе или программному обеспечению).