ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | А. Э. Зянчурин |
| должность, уч. степень, звание |  | подпись, дата |  | 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 |  |  |  | В. Ф. Губайдулин |
|  |  |  | подпись, дата |  | 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**

Ключевым документом, на основании которого ведётся разработка

является договор №5671/2023.

Реализация ведётся на основании технического задания на реализацию курсового проекта и лабораторных работ на тему «РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ/ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОБУСНОГО ВОКЗАЛА» согласованного 28.02.2023 и утверждённого 28.02.2023.

**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 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.

**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  Варианты  исп-ия | Рейс | Шаблон рейса | Автобус | Водитель | Оператор |
| 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 |  | С, U, D |  |  |  |
| Редактирование списка автобусов |  |  | С, 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

Было сформировано дерево функций изделия, которое представлено на рисунке 1.

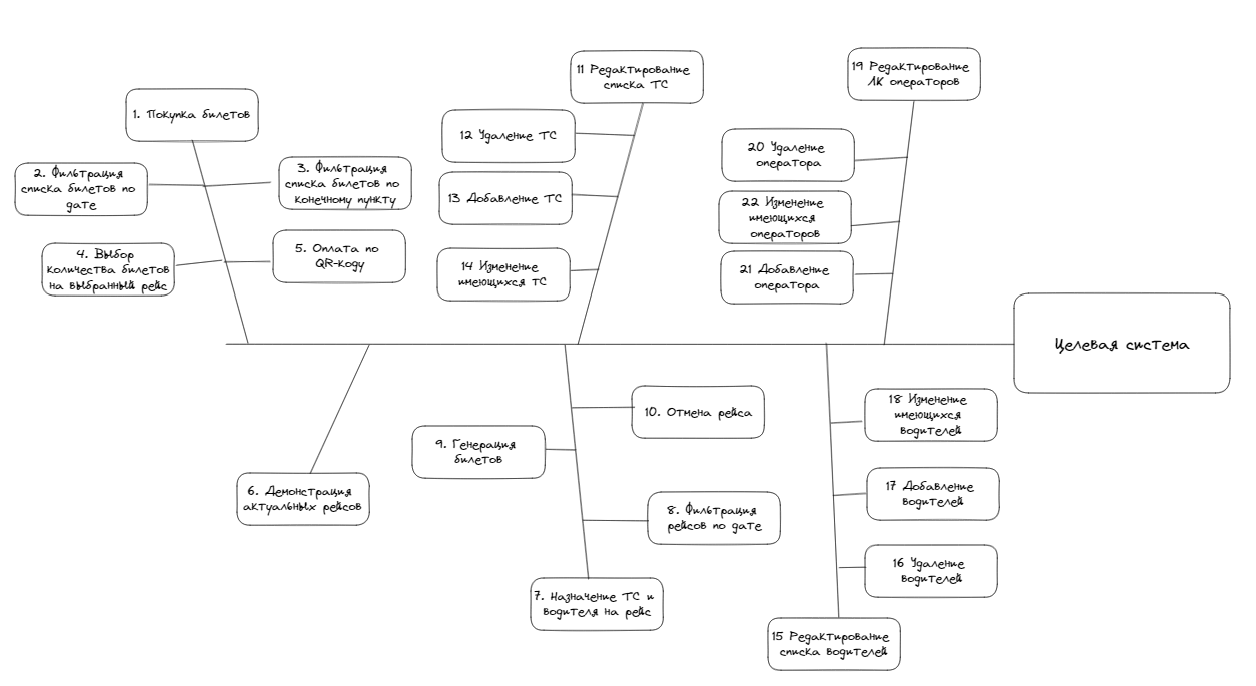


Рисунок 1 – Дерево функций изделия

**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.

Поддерживаемые браузеры: Edge, Firefox, Opera, Chrome.

**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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| Схема проверки: | - успешное добавление нового оператора; |
| Связь с элементом интерфейса: | Форма изменения операторов |
| Роли, которым доступен элемент: | Администратор |

**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 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 |

**3.2 Логическая модель**

Была реализована UML диаграмма классов, представлена на рисунке 2.

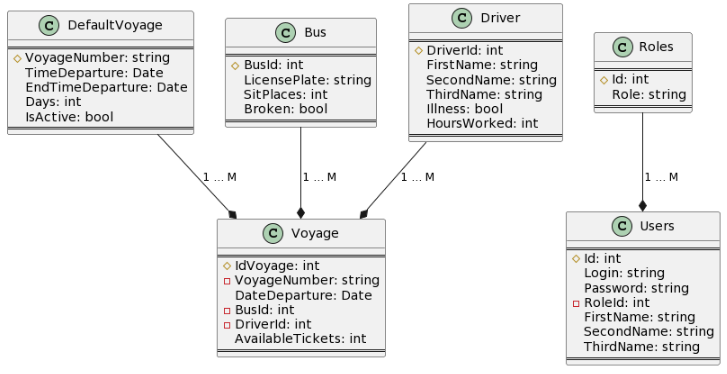


Рисунок 2 – UML-диаграмма классов

**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 логов API. Формирование файла, в котором будут писаться логи по успешности завершения сервисов внутри.
* Запись логов Front. Формирование файла, в котором будут писаться логи front-end части приложения. Записываются ошибки, которые возникли при отработке кода, отчётность о запросах в API (код, тело ответа).
* Запись логов 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 выполнится за отведённое время, он считается ошибочным и будет исправлен перформанс инженерами на стороне заказчика.

**3.4.2 Целостность**

Каждые сутки в 00:00 генерируются рейсы из шаблона рейсов на Т+2.

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

**3.4.3 Хранение**

Каждый день в 22:00 запускается бэкапирование всей БД. Бэкап файл копируется на другой сервер, предназначенный только для хранения бэкап файлов. На основном сервере файл может храниться минимум 15 дней, максимум 20 дней со дня создания, после этого он должен быть удалён. На сервере, предназначенном для бэкап файлов, файл может храниться от 3 месяцев до 6 месяцев. После этого времени также должен быть удалён.

Все логи 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**