



Федеральное государственное бюджетное образовательное
учреждение высшего образования

**"МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ)"**

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра 307 «Цифровые технологии и информационные системы»

Отчет по практическому заданию

Создание системы редактирования положения и
контроля положения товара на складе

Выполнили студенты
группы М30-321Б

«__» _____ 2021 г.

(Петина Екатерина
Викторовна)

(Добылов Баир
Юрьевич)

Принял ст.
преподаватель

«__» _____ 2021 г.

(Максимов Алексей
Николаевич)

Москва · 2021

ОГЛАВЛЕНИЕ

ПОСТАНОВКА ЗАДАЧИ	3
ТЕХНИЧЕСКОЕ ЗАДАНИЕ	3
1. <i>Введение.....</i>	3
2. <i>Основание для разработки</i>	3
3. <i>Назначение разработки.....</i>	3
<i>Создание системы складирования товара и 2D/3D визуализация складского помещения</i>	3
4. <i>Требование к программе или программному изделию.....</i>	3
<i>Персональный компьютер, погрузчик с системой позиционирования, роутер.....</i>	4
5. <i>Требования к программной документации</i>	4
6. <i>Стадии и этапы разработки</i>	4
7. <i>Порядок контроля и приёмки</i>	4
МОДЕЛЬ АРХИТЕКТУРЫ.....	5
ДИАГРАММА ПРЕЦЕДЕНТОВ	5
ОПИСАНИЕ ПРЕЦЕДЕНТОВ.....	6
ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТЕЙ	10
ОЦЕНКА МЕТОДОМ ФУНКЦИОНАЛЬНЫХ ТОЧЕК	11
МАКЕТ ИНТЕРФЕЙСА.....	12
КАРТА ЭКРАНОВ.....	14
БАЗА ДАННЫХ.....	14
1. <i>Структура базы данных</i>	14
2. <i>Таблицы</i>	15
3. <i>Наполнение таблиц</i>	16
4. <i>Выборка данных.....</i>	17

ПОСТАНОВКА ЗАДАЧИ

Целью работы является разработка программного комплекса редактирования положения и контроля положения товара на складе для системы дополненной реальности

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. Введение

Данное техническое задание определяет перечень требований к программе складского хранения и его программному обеспечению в соответствии с заданием

2. Основание для разработки

Практическое задание

3. Назначение разработки

Создание системы складирования товара и 2D/3D визуализация складского помещения

4. Требование к программе или программному изделию

4.1. Требования к функциональным характеристикам

Программа должна выполнять следующие функции:

- Редактировать и просматривать 2D план склада
- Редактировать и просматривать 3D план склада
- Размещать позиции контейнеров на складе
- Размещать позиции QR кодов на складе
- Размещать позиции стеллажей на складе
- Визуализировать положение погрузчиков
- Выдавать задание на перемещение контейнеров на погрузчик

4.2. Требование к надёжности

Не предъявляются

4.3. Условие эксплуатации

Складское помещение с достаточным уровнем освещенности, офисное помещение с автоматизированным рабочим местом

4.4. Требования к составу и параметрам технических средств

Персональный компьютер, погрузчик с системой позиционирования, роутер

4.5. Требования к информационной и программной совместимости

SQL база данных, язык программирования C++

5. Требования к программной документации

Необходимо разработать следующие документы: руководство пользователя

6. Стадии и этапы разработки

Разработка должна включать следующие стадии: определение требований, проектирование, разработка, тестирование, документирование.

7. Порядок контроля и приёмки

Результат работы принимается на последнем практическом занятии.

МОДЕЛЬ АРХИТЕКТУРЫ

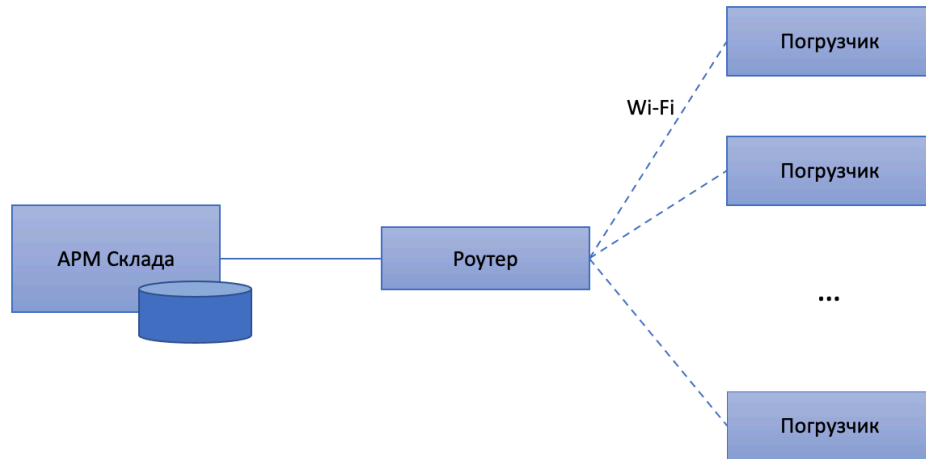


Рисунок 1. Модель архитектуры

ДИАГРАММА ПРЕЦЕДЕНТОВ

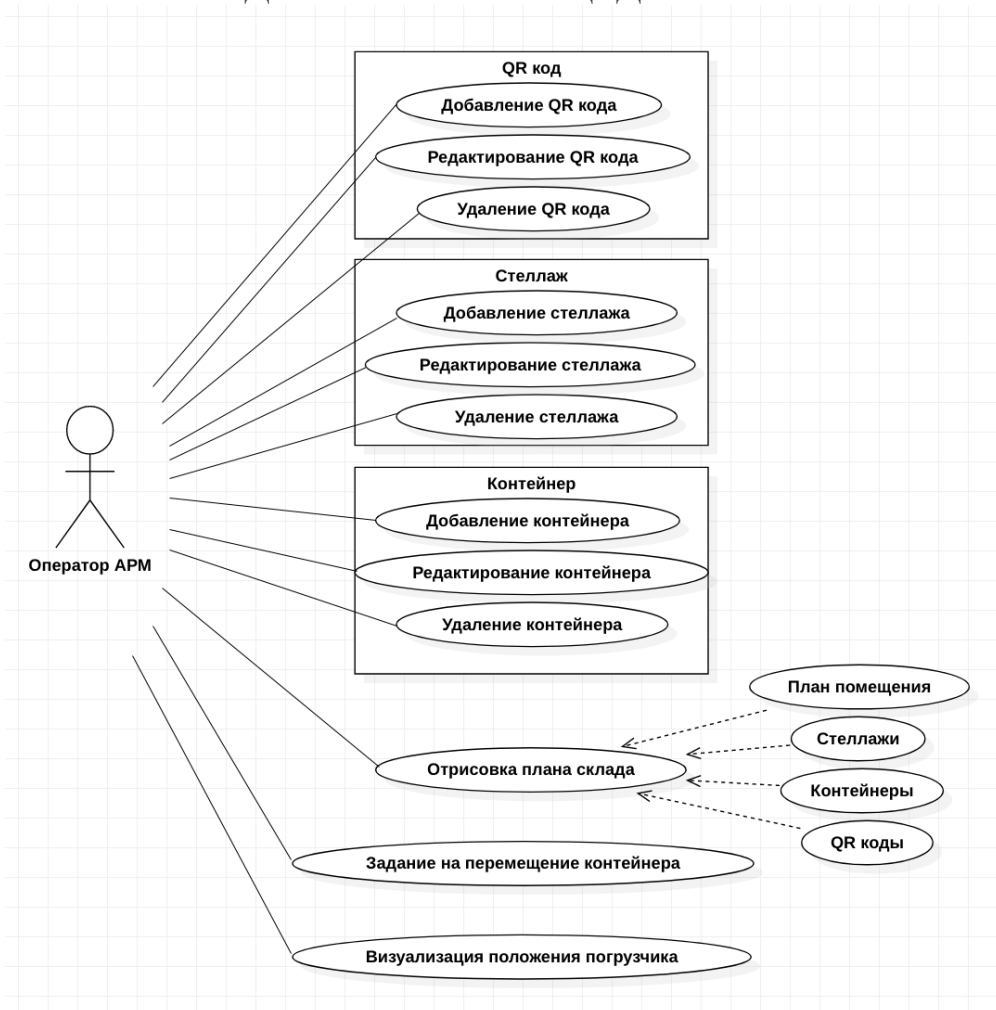


Рисунок 2. Диаграмма прецедентов

ОПИСАНИЕ ПРЕЦЕДЕНТОВ

Прецедент: Добавление QR кода (важное)

Уровень: Задача, определенная пользователем

Основной исполнитель: Оператор АРМ

Заинтересованные лица и их требования:

- оператор АРМ. Хочет добавить QR код, по которому ориентируется погрузчик

Результаты: Данные с идентификатором и координатами QR кода сохранены в базу данных

Основной успешный сценарий:

1. Оператор вводит название и координаты QR кода
2. Система формирует запрос на добавление в базу данных
3. Система извлекает из базы данные QR кодах.
4. Система обновляет список QR кодов в таблице (форме)

Прецедент: Удаление QR кода (важное)

Уровень: Задача, определенная пользователем

Основной исполнитель: Оператор АРМ

Заинтересованные лица и их требования:

- оператор АРМ. Хочет удалить QR код

Результаты: Данные QR кода удалены из базы данных

Основной успешный сценарий:

1. Оператор вводит имя и координаты контейнера
2. Система формирует запрос на добавление в базу данных
3. Система извлекает из базы данные QR кодах.
4. Система обновляет список QR кодов в таблице (форме)

Прецедент: Редактирование QR кода (необходимое)

Уровень: Задача, определенная пользователем

Основной исполнитель: Оператор АРМ

Заинтересованные лица и их требования:

- оператор АРМ. Хочет редактировать данные QR кода

Результаты: Обновленные данные с номером и положением QR кода сохранены в базу данных

Основной успешный сценарий:

1. Оператор выбирает из списка QR код, вносит изменения
2. Система формирует запрос на обновление записи в базу данных
3. Система извлекает из базы данные QR кодах.
4. Система обновляет список QR кодов в таблице (форме)

Прецедент: Добавление стеллажа (важное)
<p>Уровень: Задача, определенная пользователем</p> <p>Основной исполнитель: Оператор АРМ</p> <p>Заинтересованные лица и их требования:</p> <p>- оператор АРМ. Хочет добавить стеллаж на план склада</p> <p>Результаты: Данные с идентификатором и координатами стеллажа сохранены в базу данных</p> <p>Основной успешный сценарий:</p> <ol style="list-style-type: none"> 1. Оператор вводит название и координаты стеллажа 2. Система формирует запрос на добавление в базу данных 3. Система извлекает из базы данные о стеллажах. 4. Система обновляет список стеллажей в таблице (форме)

Прецедент: Удаление стеллажа (важное)
<p>Уровень: Задача, определенная пользователем</p> <p>Основной исполнитель: Оператор АРМ</p> <p>Заинтересованные лица и их требования:</p> <p>- оператор АРМ. Хочет удалить стеллаж</p> <p>Результаты: Данные стеллажа удалены из базы данных</p> <p>Основной успешный сценарий:</p> <ol style="list-style-type: none"> 1. Оператор выбирает стеллаж из таблицы (формы) 2. Система формирует запрос на удаление из базы данных 3. Система извлекает из базы данные стеллажей 4. Система обновляет список стеллажей в таблице (форме)

Прецедент: Редактирование стеллажа (необходимое)
<p>Уровень: Задача, определенная пользователем</p> <p>Основной исполнитель: Оператор АРМ</p> <p>Заинтересованные лица и их требования:</p> <p>- оператор АРМ. Хочет редактировать данные стеллажа</p> <p>Результаты: Обновленные данные с номером и положением стеллажа сохранены в базу данных</p> <p>Основной успешный сценарий:</p> <ol style="list-style-type: none"> 1. Оператор выбирает из списка стеллаж, вносит изменения 2. Система формирует запрос на обновление записи в базу данных 3. Система извлекает из базы данные стеллажах. 4. Система обновляет список стеллажей в таблице (форме)

Прецедент: Добавление контейнера (важное)
<p>Уровень: Задача, определенная пользователем</p> <p>Основной исполнитель: Оператор АРМ</p> <p>Заинтересованные лица и их требования:</p> <p>- оператор АРМ. Хочет добавить контейнер на план склада</p> <p>Результаты: Данные с идентификатором и координатами контейнера сохранены в базу данных</p> <p>Основной успешный сценарий:</p> <ol style="list-style-type: none"> 1. Оператор вводит название и координаты контейнера 2. Система формирует запрос на добавление в базу данных 3. Система извлекает из базы данные о контейнерах. 4. Система обновляет список контейнеров таблице (форме)

Прецедент: Удаление контейнера (важное)
<p>Уровень: Задача, определенная пользователем</p> <p>Основной исполнитель: Оператор АРМ</p> <p>Заинтересованные лица и их требования:</p> <p>- оператор АРМ. Хочет удалить контейнер</p> <p>Результаты: Данные контейнера удалены из базы данных</p> <p>Основной успешный сценарий:</p> <ol style="list-style-type: none"> 1. Оператор выбирает контейнер из таблицы (формы) 2. Система формирует запрос на удаление из базы данных 3. Система извлекает из базы данные контейнеров 4. Система обновляет список контейнеров в таблице (форме)

Прецедент: Редактирование контейнера (необходимое)
<p>Уровень: Задача, определенная пользователем</p> <p>Основной исполнитель: Оператор АРМ</p> <p>Заинтересованные лица и их требования:</p> <p>- оператор АРМ. Хочет редактировать данные контейнера</p> <p>Результаты: Обновленные данные с номером и положением контейнера сохранены в базу данных</p> <p>Основной успешный сценарий:</p> <ol style="list-style-type: none"> 1. Оператор выбирает из списка контейнер, вносит изменения 2. Система формирует запрос на обновление записи в базу данных 3. Система извлекает из базы данные контейнеров. 4. Система обновляет список контейнеров в таблице (форме)

Прецедент: Отрисовка плана склада (важное)
Уровень: Задача, определенная пользователем

Основной исполнитель: Оператор АРМ Заинтересованные лица и их требования: - оператор АРМ. Хочет просмотреть план склада Результаты: Сформирован план склада Основной успешный сценарий: 1. Оператор открывает план склада 2. Система извлекает из базы данные о стеллажах, контейнерах, QR кодах. 3. Система добавляет на план склада стеллажи, контейнеры, QR-коды

Прецедент: Выдача задания на перемещение контейнера (необходимое) Уровень: Задача, определенная пользователем Основной исполнитель: Оператор АРМ Заинтересованные лица и их требования: - оператор АРМ. Хочет выдать задание на перемещение контейнера Результаты: Данные о перемещении отправлены на погрузчик Основной успешный сценарий: 1. Оператор вводит данные контейнера 2. Система проверяет наличие контейнера в базе данных 3. Система получает координаты выбранного контейнера. 4. Система отправляет погрузчику координаты контейнера

Прецедент: Визуализация положения погрузчика (дополнительное) Уровень: Задача, определенная пользователем Основной исполнитель: Оператор АРМ Заинтересованные лица и их требования: - оператор АРМ. Хочет получить план с отображенными погрузчиками Результаты: Данные о перемещении отправлены на погрузчик Основной успешный сценарий: 1. Оператор увидеть на плане склада положение погрузчиков 2. Система получает план склада (прецедент отрисовка плана склада) 3. Система получает координаты погрузчиков. 4. Система добавляет на план склада погрузчики

Прецедент	Важность
Добавление QR кода	Важное
Удаление QR кода	Важное
Редактирование QR кода	Необходимое
Добавление стеллажа	Важное
Удаление стеллажа	Важное
Редактирование стеллажа	Необходимое

Добавление контейнера	Важное
Удаление контейнера	Важное
Редактирование контейнера	Необходимое
Отрисовка плана склада	Важное
Выдача задания на перемещение контейнера	необходимое
Визуализация положения погрузчика	дополнительное

ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Уточнить выделенное красным

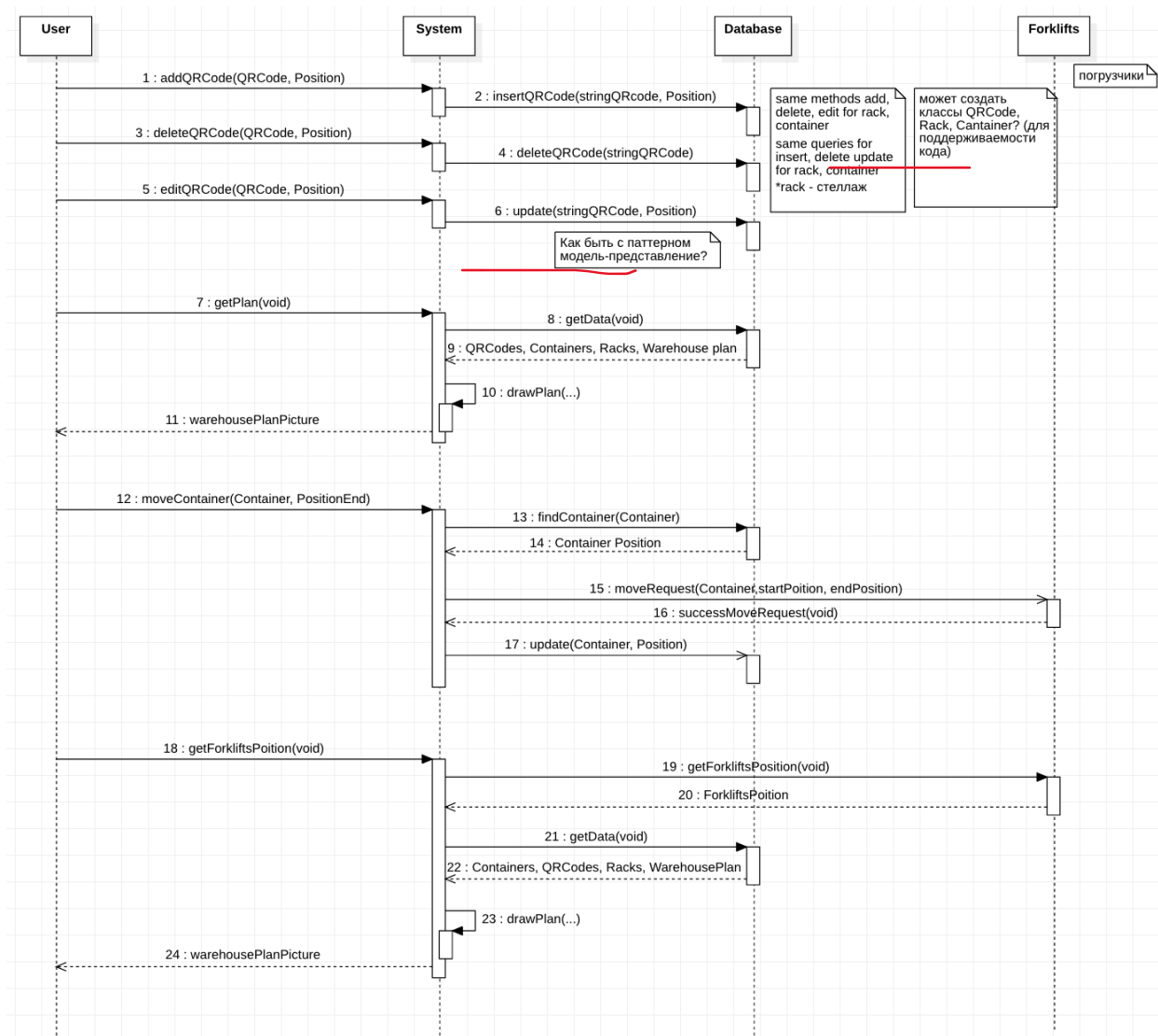


Рисунок 3. Диаграмма последовательностей

ОЦЕНКА МЕТОДОМ ФУНКЦИОНАЛЬНЫХ ТОЧЕК

Для разрабатываемой системы внешние входы, внешние выходы, внешние запросы, внутренние логические файлы и внешние логические файлы представлены в таблице рисунок 4.

Description	ILF/EIF/EI/EO/EQ	FTRs	Dets
addQRCode	EI	1	2
editQRCode	EI	1	2
deleteQRCode	EI	1	2
getPlan	ILF	1	3
moveContainer	EQ	1	1
getForkLifts	EQ	1	1
insert	EO	1	2
getData	ILF	1	3
drawPlan	EIF	1	3
findContainer	EQ	1	1
moveRequest	EQ	1	1
getForkLiftPosition	EQ	1	1

Рисунок 4.

Система имеет 3 внешних входа, 1 внешний выход, 5 внешних запросов, 2 внутренних логических файлы и 1 внешний логический файл

Подсчитаем количество функциональных точек с учетом коэффициента сложности

$$2 \times 3 + 1 \times 4 + 5 \times 3 + 2 \times 7 + 1 \times 5 = 47$$

Предварительная оценка показывает, что система имеет 47 функциональных точек. Примерное количество строк на языке C++ - 3008

МАКЕТ ИНТЕРФЕЙСА



Рисунок 5. Главное меню

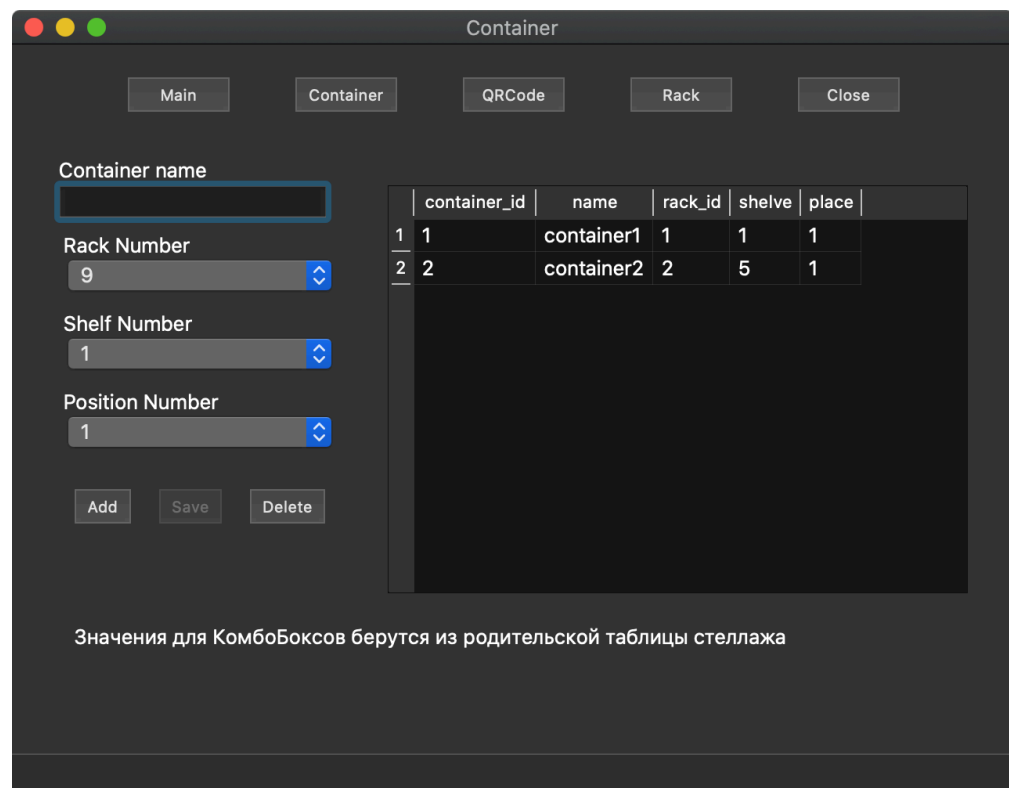


Рисунок 6. Меню контейнер

QR Code name

Horizontal position, m

Vertical position, m

Add Save Delete

	qrcode_id	name	positionX	positionY
1	1	qrcode1	110	50
2	2	qrcode2	110	90
3	7	qrcode4	310	50
4	10	qrcode5	270	170

Не понал насчет добавить 100 qr кодов timestamp - 51:00

Рисунок 7. Меню QR Код

Rack name

Position, m
horizontal vertical

Size, m
height width

Number of shelves

Number of places

Add Save Delete

	rack_id	name	positionX	positionY	height	width	shelves
1	1	rack1	50	30	50	90	10
2	2	rack2	250	30	50	90	5
3	9	rack3	350	30	50	90	3
4	11	rack4	450	30	50	90	3
5	12	rack5	100	160	50	90	3
6	13	rack6	200	160	50	90	3
7	16	rack7	400	160	50	90	3

Рисунок 8. Меню стеллаж

КАРТА ЭКРАНОВ

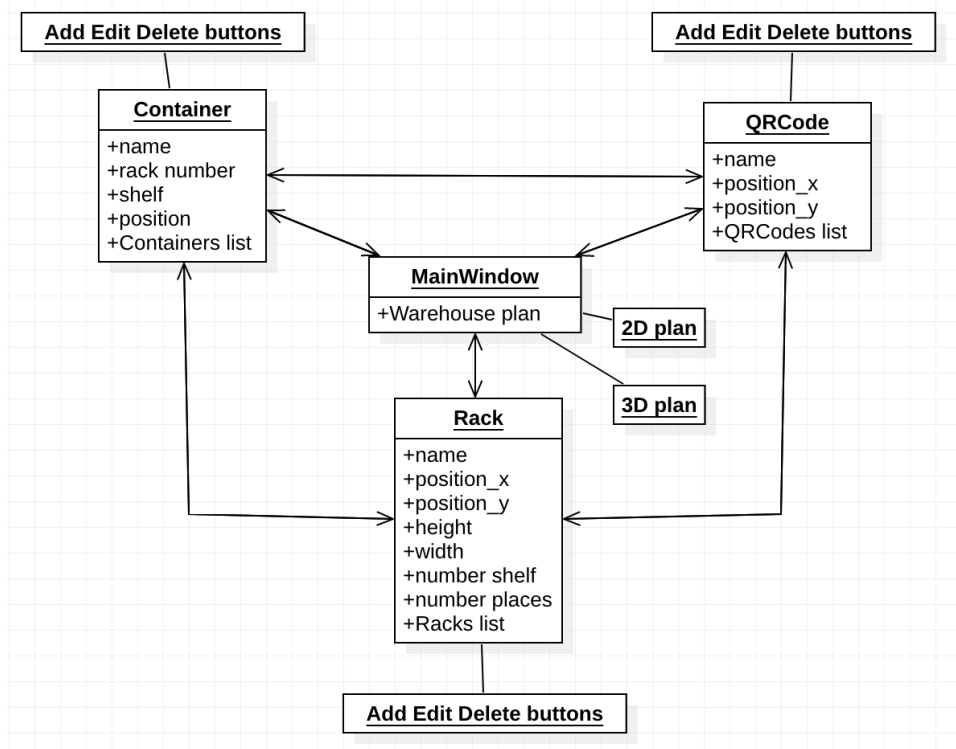


Рисунок 9. Карта экранов

БАЗА ДАННЫХ

1. Структура базы данных

Структура базы данных показана на рисунке 9 и содержит следующие таблицы: QR кодов, стеллажей и контейнеров

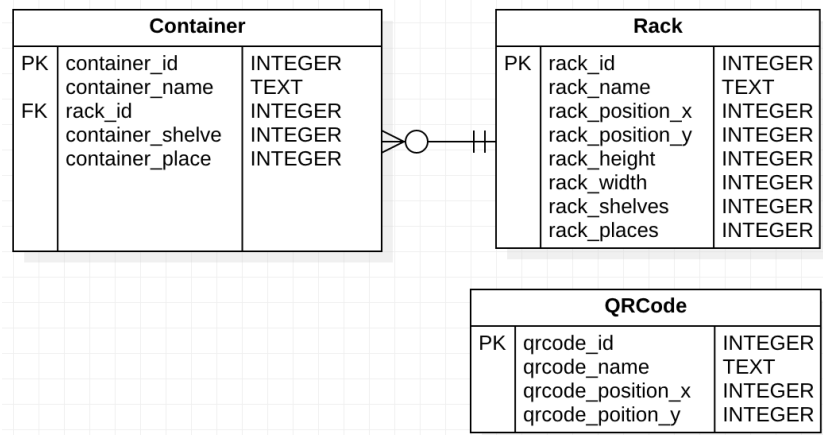


Рисунок 9. Структура базы данных

В системе используется свободная объектно-реляционная система управления базами данных PostgreSQL. Дальнейшие действия проводятся терминальном клиенте psql, работа с которым происходит интерактивно в режиме командной строки.

Создадим новую базу данных с именем warehouse:

```
postgres=# CREATE DATABASE warehouse;  
CREATE DATABASE
```

переключимся на созданную базу:

```
postgres=# \c warehouse  
You are now connected to database "warehouse" as user "*****".
```

2. Таблицы

Создадим таблицы QR кодов, стеллажей и контейнеров

```
warehouse=# CREATE TABLE QRCode(  
    qrcode_id serial PRIMARY KEY,  
    qrcode_name text,  
    qrcode_position_x integer,  
    qrcode_position_y integer);  
CREATE TABLE
```

```
warehouse=# CREATE TABLE Rack(  
    rack_id serial PRIMARY key,  
    rack_name text,  
    rack_position_x integer,  
    rack_position_y integer,  
    rack_height integer,  
    rack_width integer,  
    rack_shelves integer,
```

```
rack_places integer);  
CREATE TABLE
```

Создадим таблицу Container, включающую внешний ключ с помощью REFERENCES

```
warehouse=# CREATE TABLE Container(  
container_id serial PRIMARY KEY,  
container_name text,  
rack_id integer,  
container_shelve integer,  
container_place integer,  
FOREIGN KEY (rack_id) REFERENCES Rack(rack_id) ON DELETE  
CASCADE ON UPDATE CASCADE);  
CREATE TABLE
```

3. Наполнение таблиц

```
warehouse=# INSERT INTO rack(rack_name, rack_position_x,  
rack_position_y, rack_height, rack_width, rack_shelves, rack_places)  
VALUES ('rack1', 0, 0, 10, 10, 10, 15),  
('rack2', 50, 50, 5, 5, 5, 25);  
INSERT 0 2
```

```
warehouse=# INSERT INTO QRCode(qrcode_name, qrcode_position_x,  
qrcode_position_y)  
VALUES ('qrcode1', 10, 10),  
('qrcode2', 20, 20);  
INSERT 0 2
```


Заполним Container

```
warehouse=# INSERT INTO container(container_name, rack_id,
container_shelve, container_place)
VALUES ('container1', 1, 1, 1),
('container2', 2, 5, 1);
INSERT 0 2
```

4. Выборка данных

Выведем пять столбцов из таблицы Container:

```
warehouse=# SELECT * FROM Container
container_id | container_name | rack_id | container_shelve | container_place
-----+-----+-----+-----+-----
1 | container1 | 1 | 1 | 1
2 | container2 | 2 | 5 | 1
(2 rows)
```