|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт искусственного интеллекта (ИИИ)**

**Кафедра промышленной информатики (ПИ)**

**КУРСОВАЯ РАБОТА**

**Дисциплина «Разработка автоматизированных систем реального времени»**

**«Разработка автоматизированной системы реального времени технологического процесса производства ЖК-матриц»**

|  |  |  |  |
| --- | --- | --- | --- |
| Отчет представлен к  рассмотрению:  Студент группы КВБО-03-21 |  | (подпись) | Железняков Р.А. |
|  |  |  |  |
| Преподаватель |  | (подпись) | Зорина Н.В. |

Москва 2025 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc194571346)

[1. АНАЛИТИЧЕСКИЙ РАЗДЕЛ 4](#_Toc194571347)

[1.1 Краткая характеристика технологического процесса 4](#_Toc194571348)

[1.2 Сбор и анализ функциональных требований (UML) 10](#_Toc194571349)

[2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ 15](#_Toc194571350)

[2.1 Структурно-функциональное моделирование 15](#_Toc194571351)

[2.2 Моделирование БД 17](#_Toc194571352)

[2.3 Архитектурное моделирование 18](#_Toc194571353)

[3. РАЗРАБОТКА СИСТЕМЫ 23](#_Toc194571354)

[3.1 Выбор средств и инструментов ведения разработки 23](#_Toc194571355)

[3.2 Физическая модель базы данных 23](#_Toc194571356)

[3.3 Программная архитектура 27](#_Toc194571357)

[3.4 Разработка серверной части 28](#_Toc194571358)

[3.5 Разработка интерфейса 30](#_Toc194571359)

[4. ТЕСТИРОВАНИЕ АС 33](#_Toc194571360)

[ЗАКЛЮЧЕНИЕ 38](#_Toc194571361)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 39](#_Toc194571362)

# ВВЕДЕНИЕ

Автоматизация производства играет ключевую роль в современных промышленных предприятиях, позволяя повысить производительность, улучшить качество продукции и снизить издержки. Особенно актуально применение автоматизированных систем в высокотехнологичных отраслях, таких как производство ЖК-матриц, где точность и надежность управляемых процессов напрямую влияют на качество конечного продукта.

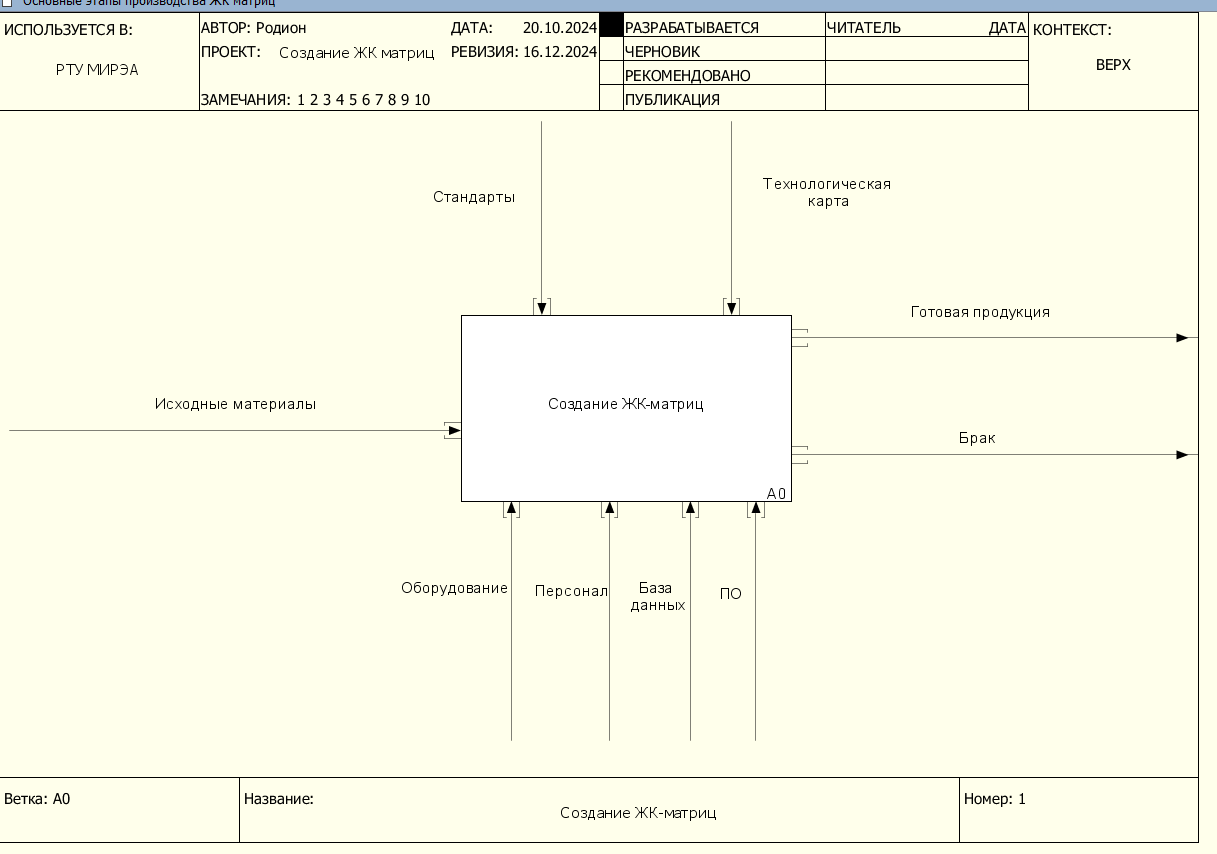
В условиях динамично развивающихся технологий, модернизация традиционных производственных процессов становится необходимым шагом для сохранения конкурентоспособности.

В процессе разработки автоматизированной системы для процесса создания ЖК-матриц необходимо выполнить подробный анализ выбранного промышленного объекта, привести параметры контроля и регулирования, выполнить подбор требуемых средств автоматизации. Построить диаграммы IDEF0 (функциональная модель) и DFD (диаграмма потоков данных) в соответствии с выбранным технологическим процессом, привести их краткое описание, логическую схему БД.

# 1. АНАЛИТИЧЕСКИЙ РАЗДЕЛ

## 1.1 Краткая характеристика технологического процесса

Технологический процесс ­­­­— создание ЖК-матриц. С использованием структурно-функционального подхода и нотации IDEF0 был проведен анализ работоспособности системы (Рисунок 1).



**Рисунок 1 — Контекстная диаграмма IDEF0**

В результате анализа выявлены необходимые для запуска процессов входные потоки:

* исходные материалы — ископаемые, с помощью которых будут созданы ЖК-матрицы. В их число входят: кварцевый песок, ацетон, полимерные смолы, фотосенсибилизаторы и др;

В качестве механизмов, позволяющих процессам выполнять проектируемые функции и задачи, выделены следующие:

* оборудование — Совокупность механизмов, машин, устройств, приборов, необходимых для производства ЖК-матриц. В число оборудования входят: муфельные печи, резаки для раскроя, установки для вакуумного напыления, фотолитографические установки, установки для реактивного ионного травления и другие;
* персонал — постоянный состав работников;
* программное обеспечение (ПО) — это совокупность программ, данных и инструкций, которые обрабатываются компьютером или другим электронным устройством для выполнения определённых задач. В число ПО входят: АСУ ТП, приложения для работы с БД и другие;
* база данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе;

Влияние на ход выполнения процессов оказывают следующие управляющие потоки:

* стандарты — нормативный документ (устанавливающий правила), отличающийся особенностями разработки, утверждения, способов использования, а также направленный на особую цель;
* технологическая карта — это стандартизированный организационно-технологический документ, содержащий необходимые сведения, инструкции для персонала, выполняющего некий технологический процесс или техническое обслуживание объекта;

В число стандартов входят:

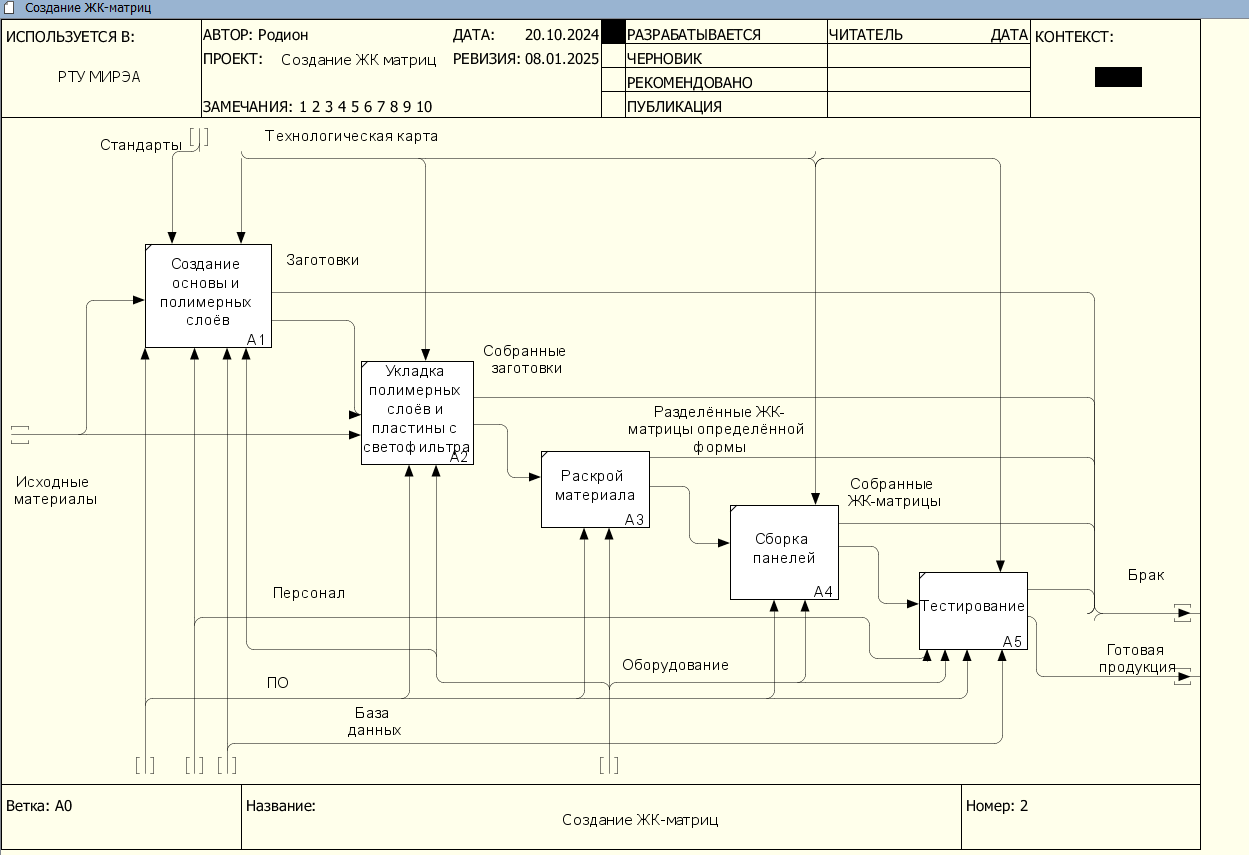
* ГОСТ 15130-86 — Стекло кварцевое оптическое. Данный стандарт позволяет создать стеклянную подложку для жидкокристаллических матриц;
* ГОСТ 24888-81 — Пластмассы, полимеры и синтетические смолы. Данный стандарт помогает создать пластиковые подложки, а также при создании фоторезисторов. Это материал, который изменяет свое сопротивление в зависимости от уровня освещенности;
* ГОСТ 2768-84 — Ацетон технический. Необходим для обработки образцов растворителем с целью удаления лишних проявленных участков. Он не производится, но благодаря содержанию ГОСТа, представляется возможным проведение анализа жидкости на пригодность;
* ГОСТ Р МЭК 61747-1-2-2017 — УСТРОЙСТВА ДИСПЛЕЙНЫЕ ЖИДКОКРИСТАЛЛИЧЕСКИЕ. Общие положения;

Технологическая карта (ТК) должна отвечать на вопросы:

* какие операции необходимо выполнять;
* в какой последовательности выполняются операции;
* с какой периодичностью необходимо выполнять операции (при повторении операции более одного раза);
* сколько потрачено времени на выполнение каждой операции;
* какие необходимы инструменты и материалы для выполнения операции.

Результатом процесса производства ЖК-матрицы являются:

* готовая продукция — ЖК-матрицы, полностью удовлетворяющие требованиям;
* брак — ЖК-матрицы, непригодные для дальнейшего использования;



**Рисунок 2 — Декомпозиция первого уровня IDEF0**

На Рисунке 2 представлена декомпозиция, отображающая основные процессы, из которых будет состоять процесс создания ЖК-матриц.

Процесс «создание основы и полимерных слоёв» представляет собой набор следующих задач:

1. Процесс начинается с подготовки стеклянного субстрата, который будет использоваться в качестве основы для ЖК-матрицы, либо с создания смеси из кварца и песка. Наиболее подходящими марками стекла для создания подложки МК-матрицы могут быть КУ-1 и КУВИ. КУ-1 обладает отличной прозрачностью и радиационной устойчивостью, а КУВИ предлагает широкий спектр прозрачности.
2. Перемешанный субстрат отправляется в муфельную печь, где плавится под высокой температурой 1200 – 1600 °C до жидкого состояния.
3. Получившиеся жидкое сырьё прокатывается под металлическими валами. В результате получаются листы стекла. После этого происходит этап охлаждение листов.
4. Параллельно происходит процесс смешивание полимерные смолы и фотосенсибилизаторов до получения однородной массы.

Процесс «укладка полимерных слоёв и пластины со светофильтром» представляет собой набор следующих задач:

1. На подложку наносятся заранее заготовленные слои фоторезистора, а также могут быть добавлены различные слои, такие как слой транзисторов (TFT), слои диэлектрика, выравнивающие слои и другие, в зависимости от конкретной технологии и типа ЖК-матрицы.
2. Укладка второго полимерного слоя: на матрицу наносится второй полимерный слой для защиты и фиксации кристаллов.
3. Также идёт процесс добавления пластины со светофильтром. Светофильтры помогают улучшить точность цветопередачи, позволяя только определенным длинам волн света проходить через матрицу.

Процесс «раскрой материала и сборка панелей» представляет собой набор следующих задач:

1. Обработка полученных заготовок ацетоном, с целью удаления лишних элементов.
2. Раскрой материала. После завершения всех этапов производства матрица подвергается процессу раскроя, где из нее вырезаются отдельные ЖК-панели.

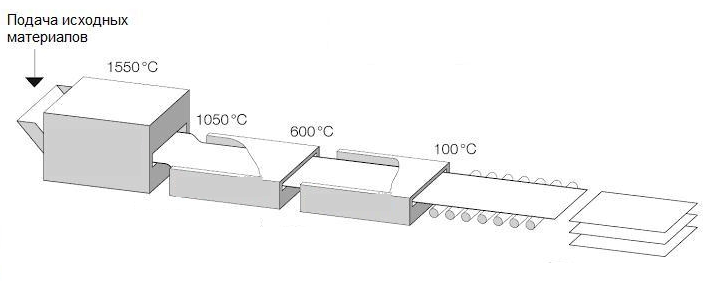
Процесс «сборка панелей» представляет собой задачу, во время выполнения которой необходимо верно расположить заготовки друг на друге, с максимальной точностью, во избежание оптических дефектов, утечки жидких кристаллов и другие дефекты.

Процесс «тестирование» представляет собой набор следующих задач:

1. Функциональное тестирование, которое включает в себе проверку всех пикселей на наличие дефектов, а также тестирование различных режимов работы.
2. Тестирование качества изображения. Среди них выделяются: оценка цветопередачи и яркости, тестирование контрастности и однородности яркости по всей поверхности дисплея.
3. Тестирование надежности, благодаря стресс-тестам для оценки долговечности дисплея.

Если после выполнения каждого процесса выявлено, что качество сырья не соответствует требованиям, то оно выходит из процесса как брак. Если качество сырья соответствует требованиям, то выполняется переход на следующий этап технологического процесса.

Для рассматриваемой технологической схемы плавления стекла (Рисунок 3) с целью создания основ для ЖК-матриц были выбраны параметры контроля и регулирования, чтобы обеспечить выходной продукт необходимого качества. Для полного контроля над процессом установлены дополнительные датчики, которые регистрируют и отображают необходимые параметры, позволяя анализировать изменения контролируемых параметров на протяжении всего рабочего цикла.



**Рисунок 3** *—* **Технологическое оборудование для создания листов стекла**

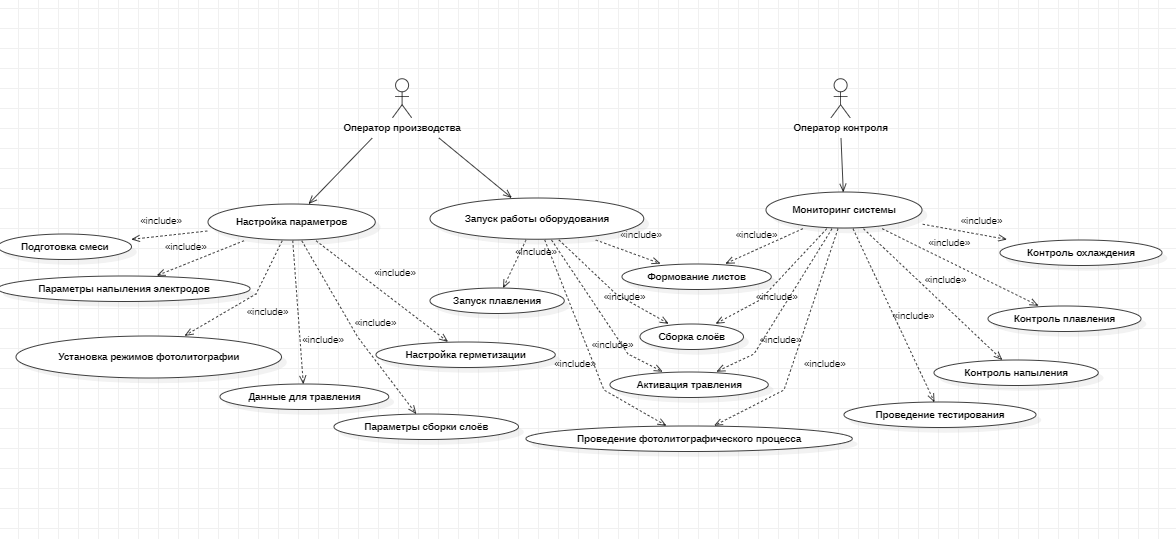
Контроль технологических процессов осуществляется с помощью измерительных приборов, которые вырабатывают сигнал о параметрах процесса. Эти приборы позволяют оперативно реагировать на изменения и поддерживать процесс выпаривания в оптимальном режиме.

*Таблица 1 — Значения технологических параметров*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Место нахождения** | **Измеряемый параметр** | **Единицы измерения** | **Номинальные значения** | **Контроль** | **Регулирование** |
| 1 | Муфель | Температура | °C | 1200 | + | + |
|  |  | Мощность | кВт | 10 | - | + |
| 2 | Стенка камеры муфеля | Закрытие двери | bool | - | + | - |
| 3 | Вентиляционная система муфельной печи | Скорость вентиляции | м/с | 1,5 | + | + |
|  |  | Фильтрация воздуха | bool | - | + | - |
|  |  | Примеси в воздухе | ppm | 500 | + | - |
| 4 | Формовочные валы | Скорость прокатки | м/мин | 10 | + | + |
|  |  | Давление при прокатке | МПа | 0,5 | + | + |
| 5 | Система охлаждения | Температура охлаждения | °C | 200 | + | + |
|  |  | Скорость охлаждения | °C/мин | 20 | + | + |

## 1.2 Сбор и анализ функциональных требований (UML)

Для успешной разработки системы необходимо определить её функциональные требования, учитывая потребности пользователей и специфику технологического процесса. Для этого будут использованы подход **Use Case** который поможет описать ключевые сценарии взаимодействия с системой.



**Рисунок 4** *—* **Функциональные требования к системе.**

В системе производства ЖК-матриц работают два ключевых оператора, каждый из которых отвечает за определённые этапы и задачи, обеспечивающие стабильность и качество производственного процесса. Оператор производства занимается подготовкой и запуском технологических операций. Он отвечает за настройку параметров, ввод необходимых значений в систему, подготовку сырья и запуск оборудования. На начальных этапах производства оператор производства вводит параметры смеси, такие как состав сырья, требуемая температура, мощность нагрева и режимы работы оборудования, что позволяет обеспечить правильное формирование стеклянной подложки. После настройки параметров он контролирует загрузку материалов, активирует муфельную печь для плавления стекломассы, следит за запуском формовочных валов, установок для нанесения тонкопленочных покрытий, фотолитографических систем и прочего оборудования, задействованного на линии. Его деятельность направлена на обеспечение корректного перехода между технологическими этапами, что является основой для получения качественного конечного продукта.

Оператор контроля осуществляет мониторинг и контроль технологического процесса на всех его этапах. Его основная задача — сбор, анализ и фиксация данных, поступающих от датчиков, а также визуальный и инструментальный контроль качества продукции. Оператор контроля регулярно проверяет соответствие измеряемых параметров нормативным значениям, отслеживает отклонения в температуре, давлении, скорости работы оборудования и других ключевых показателях. При выявлении несоответствий он фиксирует события в базе данных, инициирует корректирующие меры и обеспечивает оперативную обратную связь системе управления. Таким образом, его роль заключается в поддержании стабильности технологического процесса, обеспечении точности контроля и своевременной корректировке параметров, что способствует снижению производственных потерь и повышению качества выпускаемой продукции.

На основе данной схемы были определены функциональные требования к системе, которые представлены в Таблице 1.2.1.

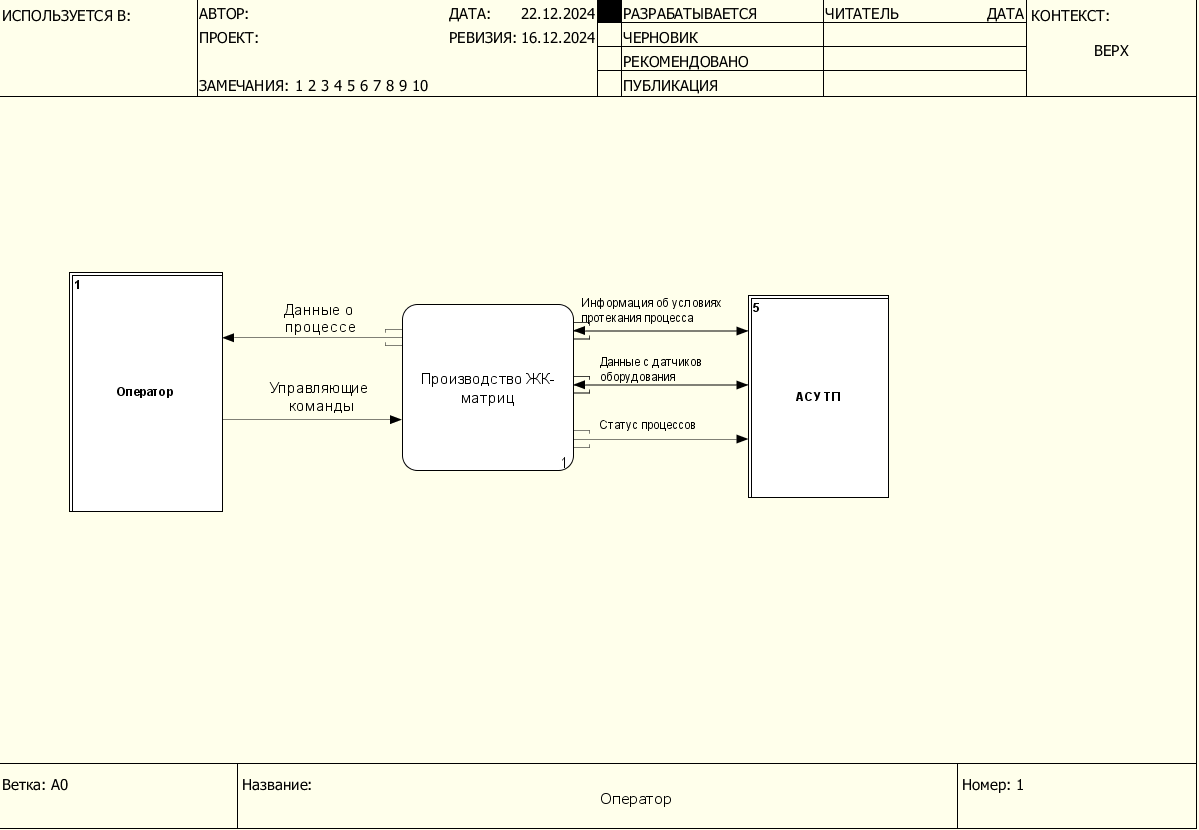
*Таблица 1.2.1 — функциональные требования*

|  |  |  |
| --- | --- | --- |
| **№** | **Функциональное требование** | **Описание** |
| 1 | Ввод и настройка параметров | Оператор производства должен вводить и корректировать параметры технологического процесса через пользовательский интерфейс. |
| 2 | Запуск технологических процессов | Инициирование запуска этапов производства. |
| 3 | Отслеживание технологического процесса | Получение данных с датчиков в режиме реального времени. Отображение результата тестирования сразу же после завершения тестирования, без обновления страницы. |
| 4 | Отображение и анализ данных | Предоставление информации о параметрах, этапах процесса, выявленных отклонениях. |
| 5 | Отслеживание результатов производства | Пользователь должен с главной страницы получить всю необходимую информации о создании ЖК-матриц. |

# ПРОЕКТИРОВАНИЕ СИСТЕМЫ

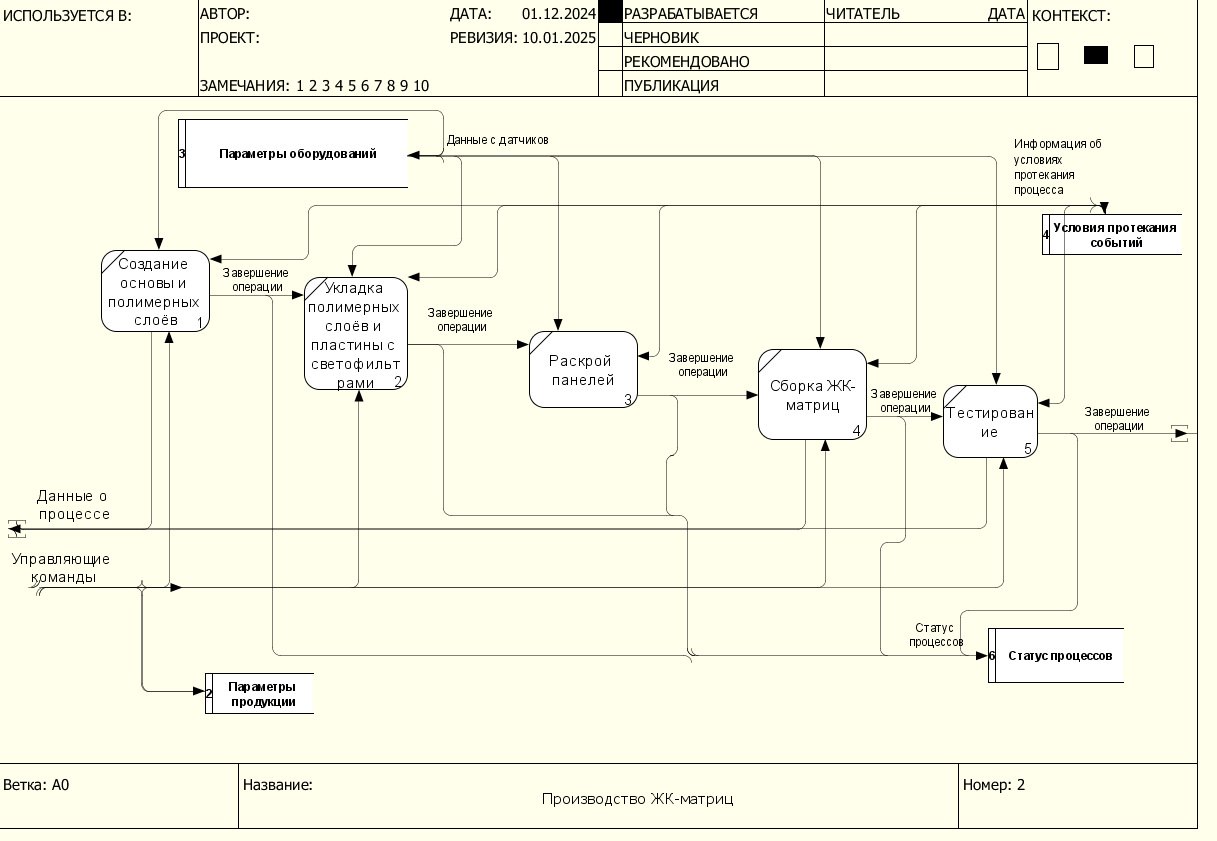
## 2.1 Структурно-функциональное моделирование

Для более детального анализа передаваемой и получаемой информации использована нотация DFD структурно-функционального подхода (Рисунок 5).



**Рисунок 5 — Контекстная диаграмма DFD**

В качестве внешних сущностей рассмотрены «оператор» и «АСУ ТП». «Оператор» представляет собой источник данных о поступающих требованиям к ЖК-матрицам, ему передается информация о протекании процессов. Сигнал об успешном завершении процесса отправляется в АСУ ТП, также в АСУ ТП хранится информация об условиях протекания процессов, данные датчиков с оборудования, информация об ошибках.



**Рисунок 6 — Декомпозиция первого уровня DFD**

В качестве хранилищ рассмотрены:

* параметры оборудований. Для хранения и предоставления информации о параметрах оборудования, при которых должны протекать процессы, данных с датчиков;
* условия протекания событий. Для хранения и предоставления информации об условиях, которые должны быть соблюдены при протекании процессов;
* параметры продукции. Предназначена для предоставления пользователю списка возможных параметров продукции, что упрощает взаимодействие с системой с помощью стандартизированных значений;
* статус процессов. Предназначен для фиксации статусов различных этапов производства. В нём также фиксируются нарушения и ошибки.

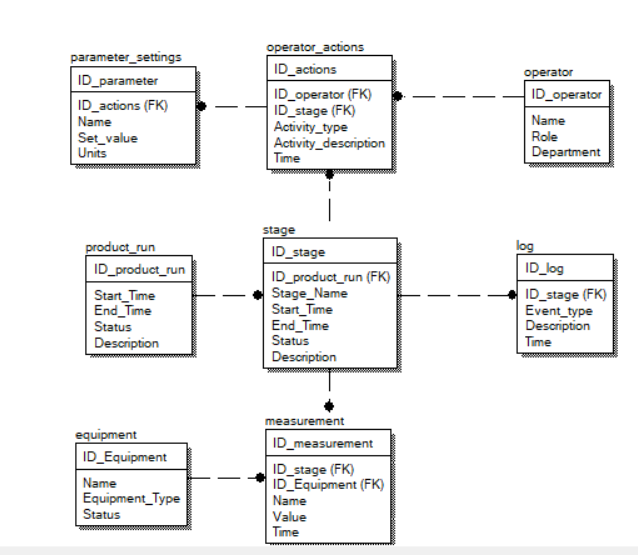
Каждый этап производства связан с предыдущим через сигнал о том, как завершился данный этап. Все они именуются как «завершение операции». Каждый сигнал также в одностороннем порядке отправляется в хранилище «Статус процессов», где и фиксируются ошибки в проведении операций.

Параллельно с этим, во время выполнения процесса идут стрелки «Данные о процесса», которые в дальнейшем идут от информационной системы к сущности «Оператор». Данные о процессах должны передаваться оператору в реальном времени, для постоянного отслеживания состояний процессов и принятия оперативных решений. В результате оператор способен отправлять управляющие команды, которые влияют на производственные процессы. Также оператор способен задать параметры продукции с помощью одноимённого хранилища.

При проведении различных операций также используются станки и другие устройства, облегчающие производственный процесс. На каждом этапе присутствуют датчики, которые фиксируют состояния изделий, состояния оборудования, точность проведения работ и т.д. Хранилище «Параметры оборудований» таким образом помогает отслеживать этапы производства. Стандартизированные условия проведения событий представлены в другом хранилище, которое служит для хранения информации о технологических параметрах, временных интервалах, необходимых для выполнения различных операций, а также стандарты и допустимые отклонения от них.

## 2.2 Моделирование БД

Спроектируем логическую схему БД, которая позволит обеспечить эффективное хранение, обработку и анализ информации, поступающей с каждого этапа технологического процесса (Рисунок 7).

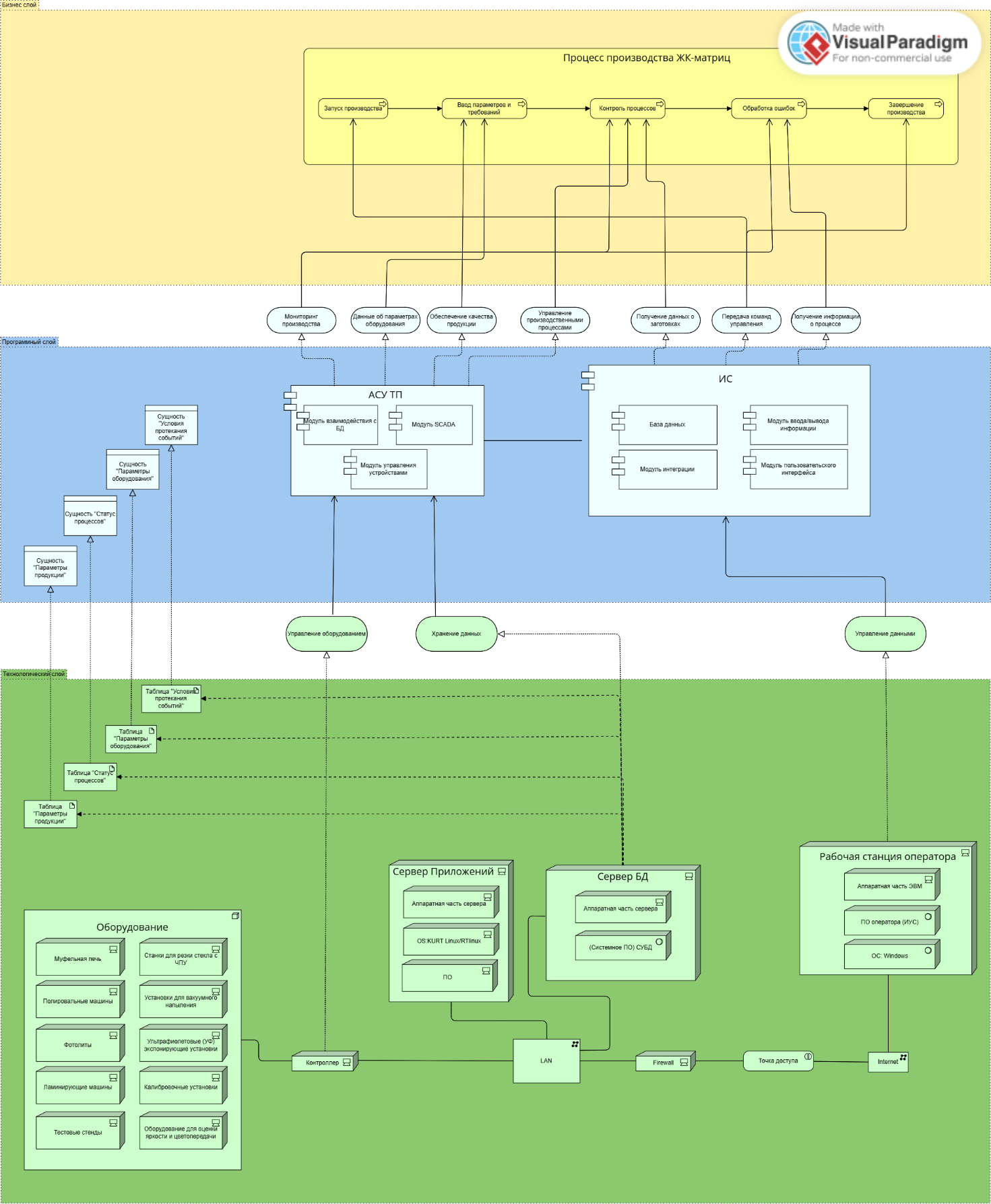


**Рисунок 7 — Логическая схема базы данных**

Основными сущностями базы данных являются данные о сотрудниках, производственных партиях, этапах производства, действиях операторов, установленных параметрах, измерениях с датчиков, оборудовании и журнале событий. Такая структура обеспечивает прозрачное отслеживание производственного цикла и позволяет оперативно реагировать на отклонения.

## 2.3 Архитектурное моделирование

Была построена диаграмма ArchiMate для производства ЖК-матриц. Диаграмма отображена на рисунке 8.



**Рисунок 5 — Диаграмма ArchiMate.**

Диаграмма состоит из 3-х слоёв: технологический, программный и слой бизнеса. Технологический слой охватывает инфраструктуру и технологии, необходимые для функционирования системы. Программный слой включает приложения и системы, которые поддерживают бизнес-процессы. Бизнес-слой описывает организационные структуры и процессы. Важно отметить, что между этими слоями существуют связи, которые показывают, как они взаимодействуют и влияют друг на друга, что является ключевым элементом в понимании архитектуры предприятия.

В технологическом слое есть несколько узлов. Самым крупным узлом является узел оборудования. Благодаря содержимому этого узла производственный процесс приходит создание ЖК-матриц. В данном узле есть следующие устройства:

1. Муфельная печь — Это устройство используется для термической обработки материалов, таких как стекло, при высоких температурах.
2. Станки для резки с ЧПУ — Эти станки управляются компьютером и позволяют точно резать различные материалы, такие как стекло и пластик, с высокой степенью точности.
3. Полировальные машины — Используются для достижения гладкой поверхности материалов, что важно для минимизации отражений и повышения качества изображения в ЖК-матрицах.
4. Установки для вакуумного напыления — Эти устройства применяются для нанесения тонких пленок на поверхности подложек.
5. Фотолиты — Это устройства, используемые для создания фотошаблонов на подложках. Они обеспечивают точное нанесение светочувствительных материалов, которые затем будут подвергаться экспонированию и проявлению для формирования необходимых структур.
6. Ультрафиолетовые экспонирующие установки — Эти установки используются для облучения фотосенсибилизированных слоев, что позволяет создавать микроструктуры на подложках.
7. Ламинирующие машины — Эти устройства предназначены для соединения нескольких слоев материалов в единое целое.
8. Калибровочные установки — Используются для проверки и настройки параметров готовых изделий.
9. Тестовые стенды — Эти устройства предназначены для проведения различных тестов на готовых продуктах, таких как проверка функциональности, надежности и качества изображения.
10. Оборудования для оценки яркости и цветопередачи — Также используются для проведения тестирования устройств на качество яркости ЖК-матриц.

Общение с оборудование происходит через контроллер. Контроллер соединён с Local Area Network (LAN в дальнейшем). LAN соединён с сервером приложений и сервером БД. Сервер приложений необходим для сбора и надёжного архивирования данных со всех поступающих устройств. Он состоит из аппаратной части сервера, которая содержит в себе процессор, ОЗУ, HDD, блок питания и другие элементы. Также на сервере есть операционная система — KURT Linux. Это расширение для работы ядра в реальном времени. Также на сервере будет расположено программное обеспечение, которое реализует работу АСУ ТП и ИС. Сервер БД необходим для хранения данных о произведённых ЖК-матрицах. Также он хранит в себе значения параметров оборудования и продукции. Параметры оборудования содержат информации о работе отдельных элементов оборудования. Параметры продукции несут в себе информацию о том, какая продукция должна выпускаться по итогу работы производственного процесса. На сервер БД также должна быть установлена СУБД. Также к LAN через Firewall, точка доступа и Internet, подключена рабочая станция оператора. Рабочая станция оператора содержит в себе аппаратную часть ЭВМ, отличную от сервера меньшими мощностями. На рабочую станцию ЭВМ установлена ИС — система, занимающаяся управлением производственным процессом. Через неё оператор осуществляет управление производством ЖК-матриц.

Следующий слой — программный. На него приходят абстракции сущностей тех таблиц, которые были получены с сервера БД. Также в программном слое расположены АСУ ТП и информационная система. Внутри АСУ ТП расположена SCADA-система, которая предназначена для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. Модуль управления устройствами занимается внесением корректировок в работу устройств, с целью получения необходимой продукции. Информационная система с элементами управления — ПО, предназначенная для введения, управления и отображения информации. При этом данное ПО должно верно отрисовать пользовательский интерфейс, который был бы понятен для оператора. Поскольку идёт управление информацией через взаимодействие с АСУ ТП, необходима базы данных.

Последний слой — бизнес-слой. Он отображает функции, которые выполняет пользователь. Основная функция — процесс производства ЖК-матриц. Оператор в данном процессе может выполнять следующее: запуск производства, ввод параметров и требований, контроль процессов, обработка ошибок и завершение производства.

# РАЗРАБОТКА СИСТЕМЫ

## 3.1 Выбор средств и инструментов ведения разработки

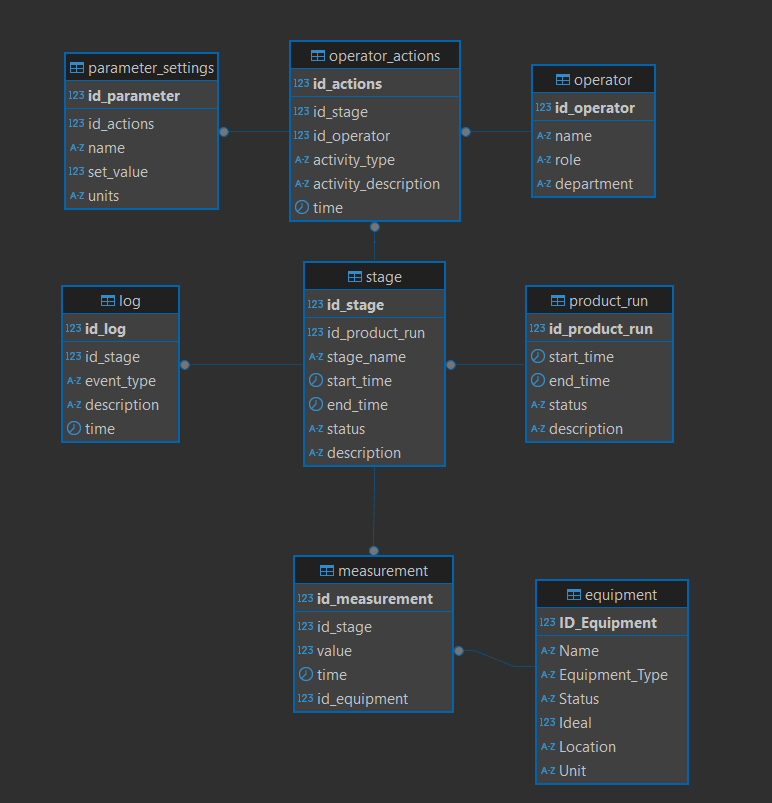
Проект будет реализован по принципу клиент-серверной архитектуры. Это означает, что на нашем сервере будет реализована основная логика. Пользователи системы будут подключаться к серверу через веб-приложение. Используемый язык программирование — Python. Он был выбран в связи с его богатой экосистемой библиотек, а также высокой производительностью при работе с асинхронными задачами. Данный язык программирования интегрируется с современными технологиями, такими как FastAPI и Uvicorn. Это позволяет создавать масштабируемые серверы с поддержкой WebSocket. Благодаря этой технологии реализовывается поддержка реального времени в приложении. Кроме того, наличие таких инструментов, как Pydantic для валидации данных и SQLAlchemy для работы с базами данных, значительно ускоряет разработку и обеспечивает высокий уровень контроля над процессами, что особенно важно для автоматизированной системы производства ЖК-матриц.

Пользовательский интерфейс будет реализован с помощью библиотеки React. React позволит создавать модульный и легко масштабируемый интерфейс, который динамически обновляется при изменении данных, получаемых с сервера через WebSocket.

Для хранения информации о производстве будет использоваться реляционная база данных. Выбор пал на базу данных PostgreSQL. Данный выбор обоснован тем, что PostgreSQL обеспечивает высокую надежность, масштабируемость и безопасность хранения данных.

## 3.2 Физическая модель базы данных

На рисунке 6 представлена физическая модель базы данных.



**Рисунок 6 — Физическая модель базы данных**

В данной базе данных представлены следующие таблицы:

1. Таблица “operator”: служит для хранения данных о рабочем персонале. Поле “id\_operator” — первичный ключ, целочисленного типа данных. Он не может быть пустым, и в нём заложена параметр автоинкремента. Следующее поле называется “name”. Тип данных — varchar(64). Это означает, что строка будет переменной длины, а её максимальное значение – 64 символа. Поле “role” служит для обозначения должности работника. Тип данных — varchar(20). Поле “department” служит для того, чтобы отобразить цех, на котором работает сотрудник. Тип данных — varchar(20).
2. Таблица “equipment”: Данная таблица служит для хранения данных об оборудовании, используемом в технологическом процессе. Поле “ID\_Equipment” является первичным ключом, целочисленного типа данных, не может быть пустым и использует автоинкремент. Поле “name” имеет тип данных varchar(64), что означает переменную строку с максимально допустимой длиной 64 символа. Поле “equipment\_type” также имеет тип varchar(64) и используется для указания типа оборудования. Поле “status” имеет тип varchar(20) и предназначено для хранения текущего статуса оборудования.
3. Таблица “log”: Эта таблица предназначена для хранения записей журнала событий, происходящих в процессе производства. Поле “ID\_log” — первичный ключ, целочисленного типа с автоинкрементом, не может быть пустым. Поле “ID\_stage” имеет тип INTEGER и служит внешним ключом, связывающим данную запись с таблицей “stage” по полю “ID\_stage”. Поле “event\_type” имеет тип varchar(64) и хранит тип события (например, предупреждение или ошибка). Поле “description” имеет тип varchar(256) и может содержать описание события; допускается значение NULL. Поле “time” имеет тип TIMESTAMP и не может быть пустым, оно фиксирует время регистрации события.
4. Таблица “measurement”: Таблица “measurement” используется для хранения данных измерений, получаемых с датчиков во время технологического процесса. Поле “ID\_measurement” является первичным ключом, целочисленного типа с автоинкрементом, не может быть пустым. Поле “ID\_stage” имеет тип INTEGER, не может быть пустым и является внешним ключом, связывающим измерение с соответствующим этапом производства в таблице “stage” (поле “ID\_stage”). Поле “name” имеет тип varchar(64) и используется для хранения названия измеряемого параметра. Поле “value” имеет тип INTEGER и содержит числовое значение измерения. Поле “time” имеет тип TIMESTAMP, фиксируя время измерения, и не может быть пустым. Поле “ID\_equipment” имеет тип INTEGER, не допускает пустых значений и является внешним ключом, ссылающимся на таблицу “equipment” по полю “ID\_equipment”.
5. Таблица “operator\_actions”: Эта таблица хранит информацию о действиях, выполненных операторами в процессе производства. Поле “ID\_actions” является первичным ключом, целочисленного типа с автоинкрементом, не может быть пустым. Поле “ID\_stage” имеет тип INTEGER, не допускает пустых значений и является внешним ключом, связывающим данную запись с таблицей “stage” (поле “ID\_stage”). Поле “ID\_operator” имеет тип INTEGER, не может быть пустым и является внешним ключом, указывающим на таблицу “operator” по полю “ID\_operator”. Поле “activity\_type” имеет тип varchar(20) и предназначено для указания типа выполняемой деятельности. Поле “activity\_description” имеет тип varchar(256) и содержит описание действия оператора. Поле “Time” имеет тип TIMESTAMP, фиксируя время выполнения действия, и не может быть пустым.
6. Таблица “parameter\_settings”: Таблица “parameter\_settings” предназначена для хранения параметров, установленных операторами во время выполнения действий. Поле “ID\_parameter” является первичным ключом, целочисленного типа с автоинкрементом, не допускает пустых значений. Поле “ID\_actions” имеет тип INTEGER, не может быть пустым и является внешним ключом, связывающим запись с таблицей “operator\_actions” по полю “ID\_actions”. Поле “name” имеет тип varchar(64) и используется для хранения названия настраиваемого параметра. Поле “set\_value” имеет тип FLOAT и содержит значение, установленное оператором. Поле “units” имеет тип varchar(20) и предназначено для хранения единиц измерения данного параметра.
7. Таблица “product\_run”: Эта таблица хранит сведения о производственных партиях или запусках. Поле “ID\_product\_run” является первичным ключом, целочисленного типа с автоинкрементом, не допускает пустых значений. Поле “start\_time” имеет тип TIMESTAMP и фиксирует время начала производственного запуска. Поле “end\_time” также имеет тип TIMESTAMP, фиксируя время завершения партии. Поле “status” имеет тип varchar(20) и хранит статус производственного запуска (например, “в процессе”, “завершено”). Поле “description” имеет тип varchar(256) и может содержать дополнительные комментарии, допускается значение NULL.
8. Таблица “stage”: Таблица “stage” предназначена для хранения информации об этапах производственного процесса, входящих в состав производственных запусков. Поле “ID\_stage” является первичным ключом, целочисленного типа с автоинкрементом, не может быть пустым. Поле “ID\_product\_run” имеет тип INTEGER, не допускает пустых значений и является внешним ключом, связывающим данный этап с таблицей “product\_run” (поле “ID\_product\_run”). Поле “stage\_name” имеет тип varchar(20) и используется для хранения названия этапа. Поле “start\_time” имеет тип TIMESTAMP и фиксирует время начала этапа, не может быть пустым. Поле “end\_time” также имеет тип TIMESTAMP, фиксируя время завершения этапа, не допускает пустых значений. Поле “status” имеет тип varchar(20) и предназначено для хранения статуса этапа (например, “в процессе”, “завершено”). Поле “description” имеет тип varchar(256) и может содержать дополнительное описание этапа; допускается значение NULL.

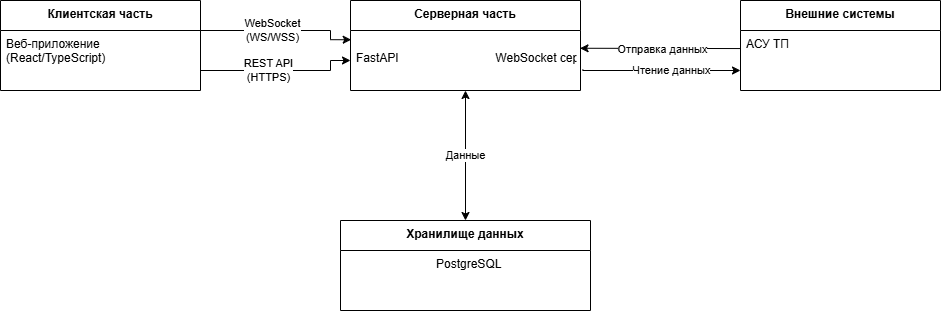
На листинге 3.1 представлен пример SQL запроса для создания таблицы, на основе описания таблицы “measurement”.

*Листинг 3.1 — Пример SQL запроса для создания таблицы в БД*

|  |
| --- |
| *CREATE TABLE measurement (*  *id\_measurement SERIAL PRIMARY KEY,*  *id\_stage INT NOT NULL REFERENCES stage(id\_stage),*  *value INT NOT NULL,*  *time TIMESTAMP NOT NULL,*  *id\_equipment INT NOT NULL REFERENCES equipment(id\_equipment)*  *);* |

## 3.3 Программная архитектура

На рисунке 7 представлена клиент-серверная архитектура.



**Рисунок 8 — Клиент-серверная архитектура ПО**

Клиентская часть включает в себя веб-интерфейс, который взаимодействует с сервером через HTTP-запросы и WebSocket-подключение. Серверная часть содержит REST API и WebSocket-сервер, которые обеспечивают обработку запросов и событий в реальном времени. Основная бизнес-логика сосредоточена на сервере и управляет взаимодействием между клиентом, внешними системами и хранилищем данных.

Для работы с внешними системами, такими как АСУ ТП, предусмотрены механизмы чтения и отправки данных. В качестве основного хранилища данных используется база данных PostgreSQL, с которой сервер взаимодействует через бизнес-логику.

## Разработка серверной части

Серверная часть представляет собой приложение, в которое встроено эмулирование получения данных путём их вычисления на основе идеальных значений. Клиент отправляет сообщение через WebSocket, которое запускает эмуляцию, которую в дальнейшем записывает в БД. Для этого необходимо сделать так, чтобы в бесконечном цикле сервер ожидал входящие текстовые сообщения с клиента. Для организации двунаправленного общения используется WebSocket-эндпоинт, который настроен следующим образом (Листинг 3.2):

*Листинг 3.2 — Двунаправленное общение*

|  |
| --- |
| *@router.websocket("/ws") async def websocket\_endpoint(websocket: WebSocket):  await ws\_manager.connect(websocket)  try:  while True:  data = await websocket.receive\_text()  except WebSocketDisconnect:  print("Клиент отключился нормально")  except Exception as e:  print(f"Неожиданная ошибка: {e}")  finally:  ws\_manager.disconnect(websocket)  await websocket.close()* |

Подключение происходит путём установки соединения, во время которого вызывается метод ws\_manager.connect(websocket), который регистрирует новое WebSocket-соединение и добавляет его в список активных соединений.

Как только функция эмулирования данных отработает, по созданному WebSocket-соединению отправляется соответствующее сообщение, которое сигнализирует об успешности проведённой операции, параллельно записывая в таблицу лог. Часть кода представлено на Листинге 3.3

*Листинг 3.3 — Сообщение для клиента*

|  |
| --- |
| *async with session.begin():  stage = await session.get(StageModel, stage\_id)  if stage:  stage.status = "completed"  stage.end\_time = datetime.now(timezone.utc).replace(tzinfo=None),   end\_log = LogModel(  id\_stage=stage\_id,  event\_type="STAGE\_COMPLETED",  description=f"Stage {stage\_name} completed successfully",  time=datetime.now(timezone.utc).replace(tzinfo=None),  level="INFO"  )  session.add(end\_log)   await ws\_manager.broadcast({  "status": "completed",  "stage\_id": stage\_id,  "stage\_name": stage\_name  })   await commit\_with\_retry(session)* |

Запись данных на сервере осуществляется с помощью ORM SQLAlchemy. При этом создаётся сессия базы данных, затем с помощью методов add() и commit() объекты (например, модели измерений) вставляются в базу данных, а транзакция завершается вызовом commit(). Это позволяет серверу работать с базой данных через объектно-реляционное отображение. Пример создания модели представлен на Листинге 3.4.

*Листинг 3.4 — Модель таблицы «stage»*

|  |
| --- |
| *class Stage(Base):  \_\_tablename\_\_ = "stage"   id\_stage = Column("id\_stage", Integer, primary\_key=True, autoincrement=True)  id\_product\_run = Column("id\_product\_run", Integer, ForeignKey("product\_run.id\_product\_run"), nullable=False)  stage\_name = Column("stage\_name", String(20), nullable=False)  start\_time = Column(DateTime(timezone=False), nullable=False)   end\_time = Column(DateTime(timezone=False), nullable=True)  status = Column("status", String(20), nullable=False)  description = Column("description", String(256), nullable=True)   product\_run = relationship("ProductionRun", back\_populates="stages")* |

## 3.5 Разработка интерфейса

На рисунке 9 представлена главная страница приложения.

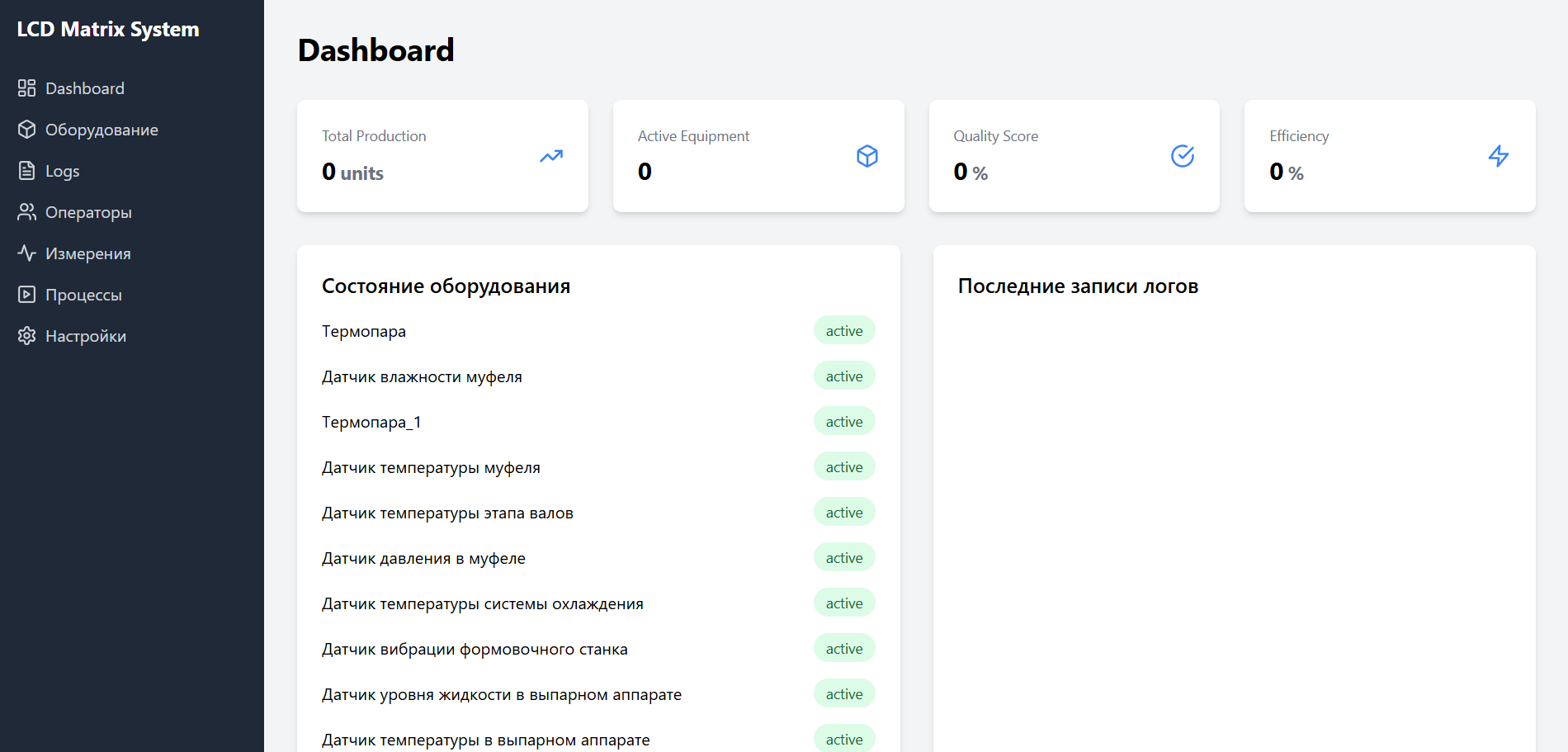
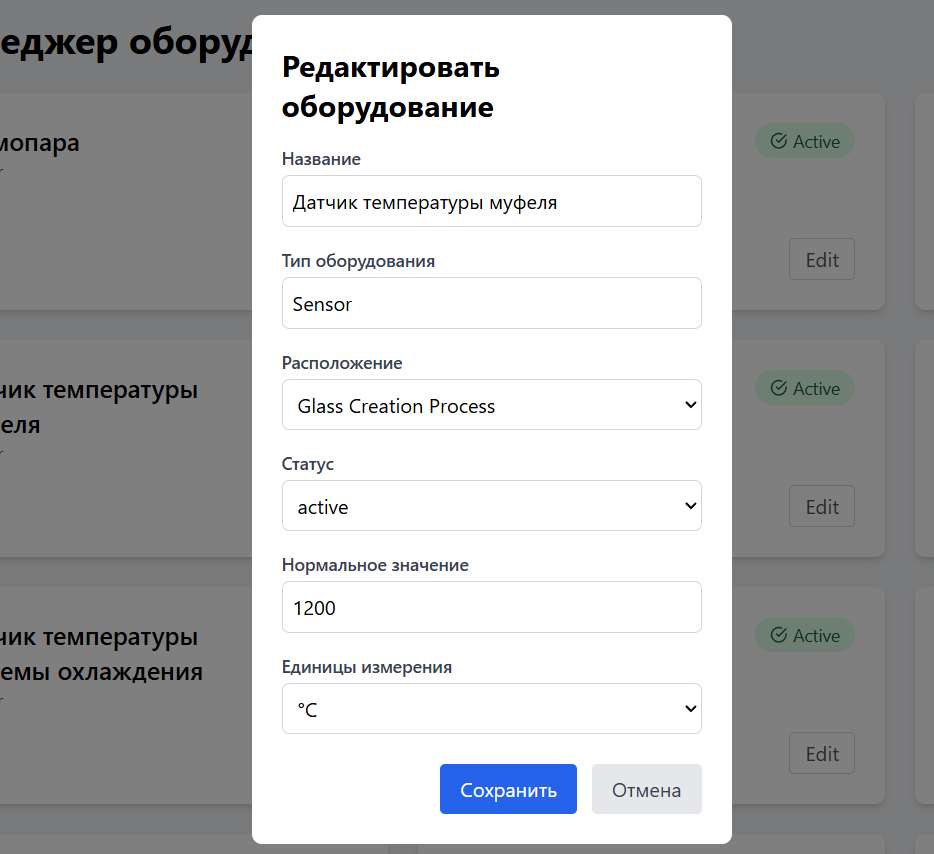


Рисунок 9 — Интерфейс ПО

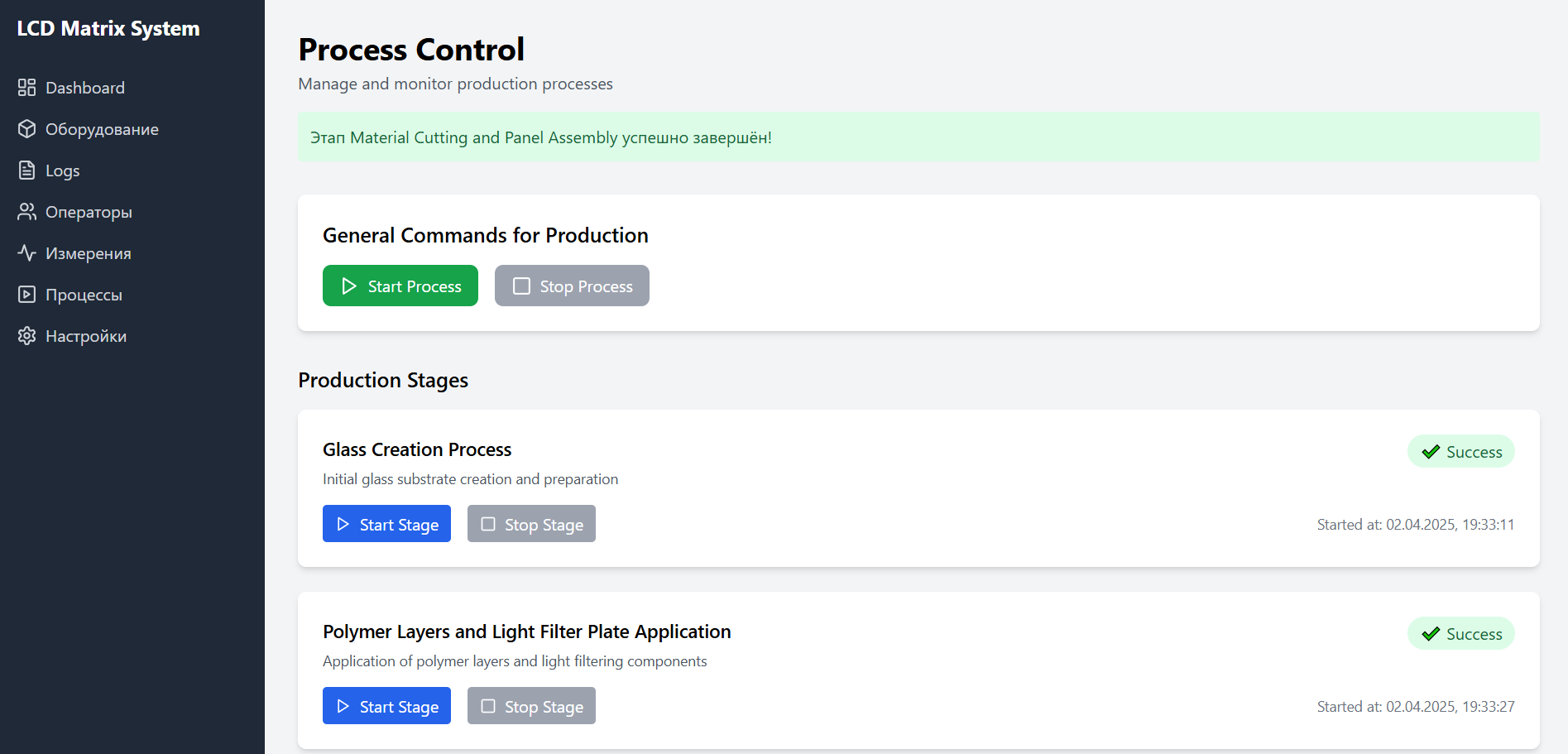
Для разработки фронтенд-части были использованы такие технологии, как HTML5/CSS3/JavsScript, в качестве основы интерфейса. React был использован для создания адаптивного и современного дизайна. В левой части интерфейса расположена панель навигации, где отображается название приложения, а также следующие страницы: Dashboard, Оборудование, Logs, Операторы, Измерения, Процессы и Настройки. Каждая из страниц связанна с соответствующим эндпоинтом. Основными же является раздел Оборудование, Процессы и Измерения.

На странице с оборудованием идёт перечисление добавленного пользователем датчиков, станков и других инструментов, необходимых для производства ЖК-матриц. Также у каждого из них есть индикаторы и возможность определения в соответствующий производственный процесс. На рисунке 10 представлено окно, позволяющее заполнить сведенья об устройстве.



**Рисунок 10 — Окно настройки параметров оборудования**

В разделе Процессы представлена страница, позволяющая запустить производственные технологические процессы. Страница представлена на рисунке 11.



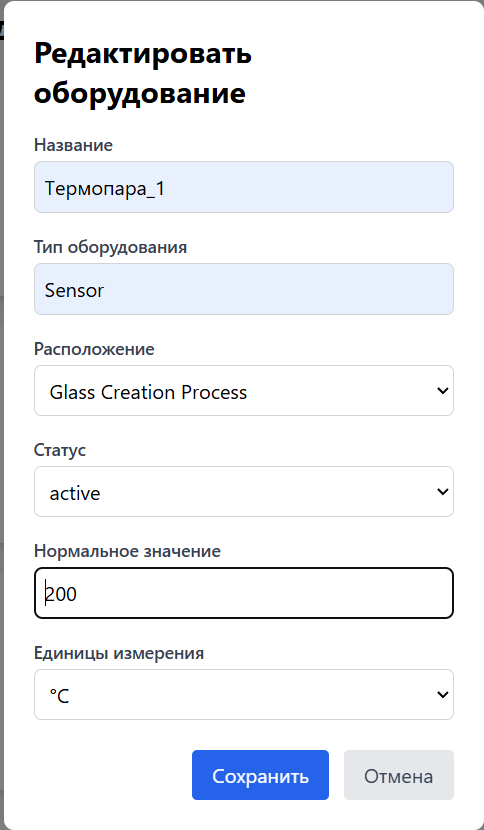
**Рисунок 11 — Панель управления процессами**

# ТЕСТИРОВАНИЕ АС

Проведём функциональное тестирование на основе требований, которые были выявлены в Use case диаграмме. Разберём каждый пункт по отдельности:

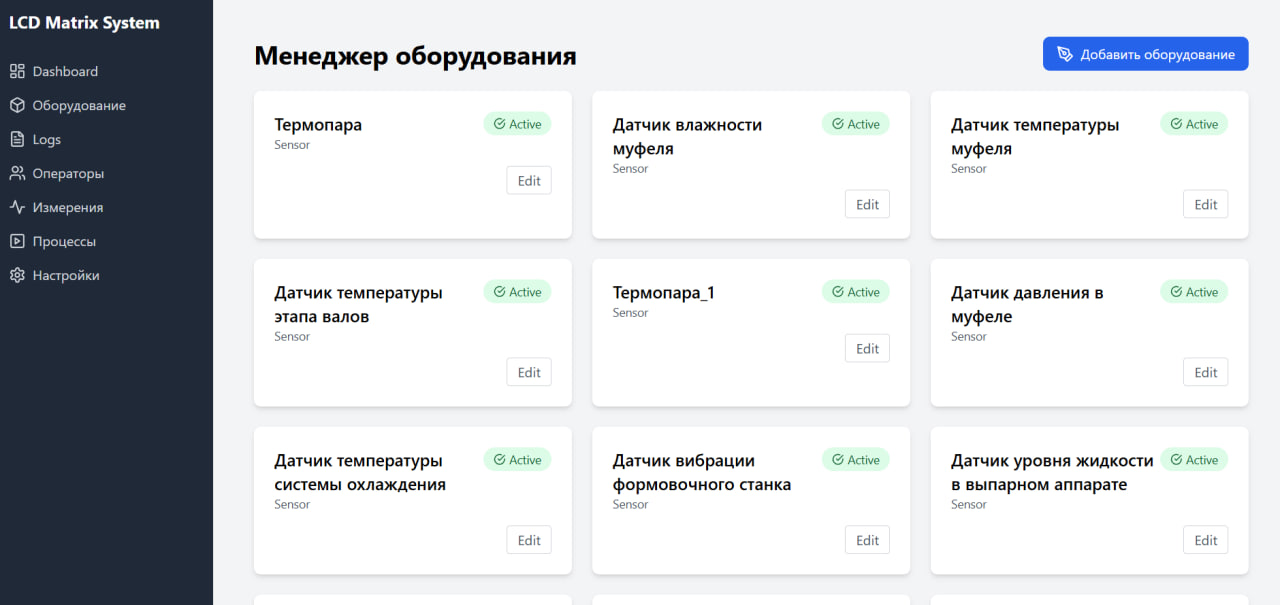
1. Ввод и настройка параметров. Оператор производства должен вводить и корректировать параметры технологического процесса через пользовательский интерфейс.

Для реализации данной функции была создана страница с оборудованием. На данной странице есть кнопка «добавить оборудование», которое позволяет пользователю добавить новое оборудование. После этого открывается отдельное окно, которое представлено на рисунке 4.1.

****

**Рисунок 4.1 — Добавление оборудования**

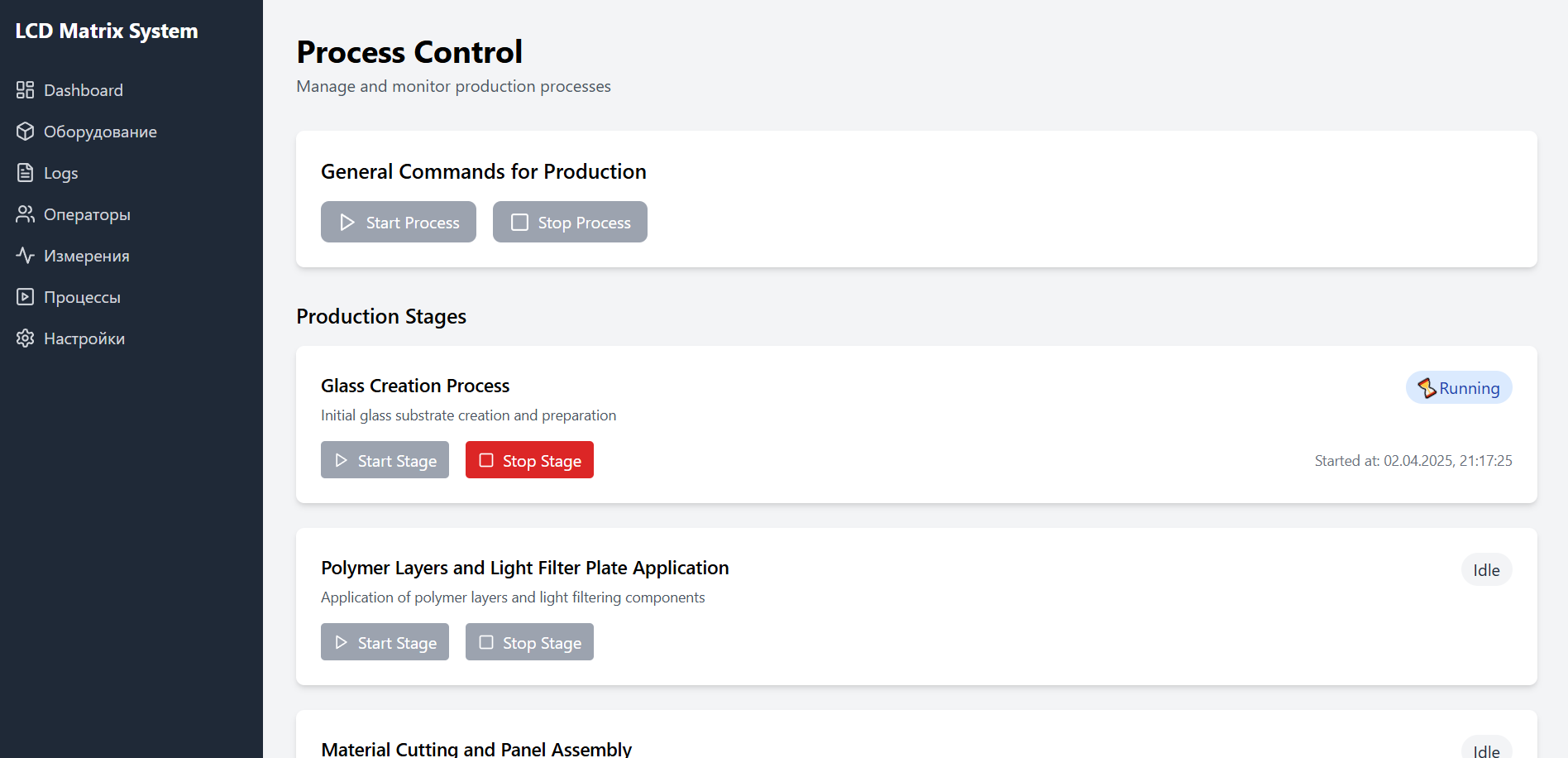
После внесения оборудования в систему, оно отображается на общей странице менеджера оборудования. Результат представлен на рисунке 4.2.



**Рисунок 4.2 — Активный модуль, добавленный с помощью специального окна**

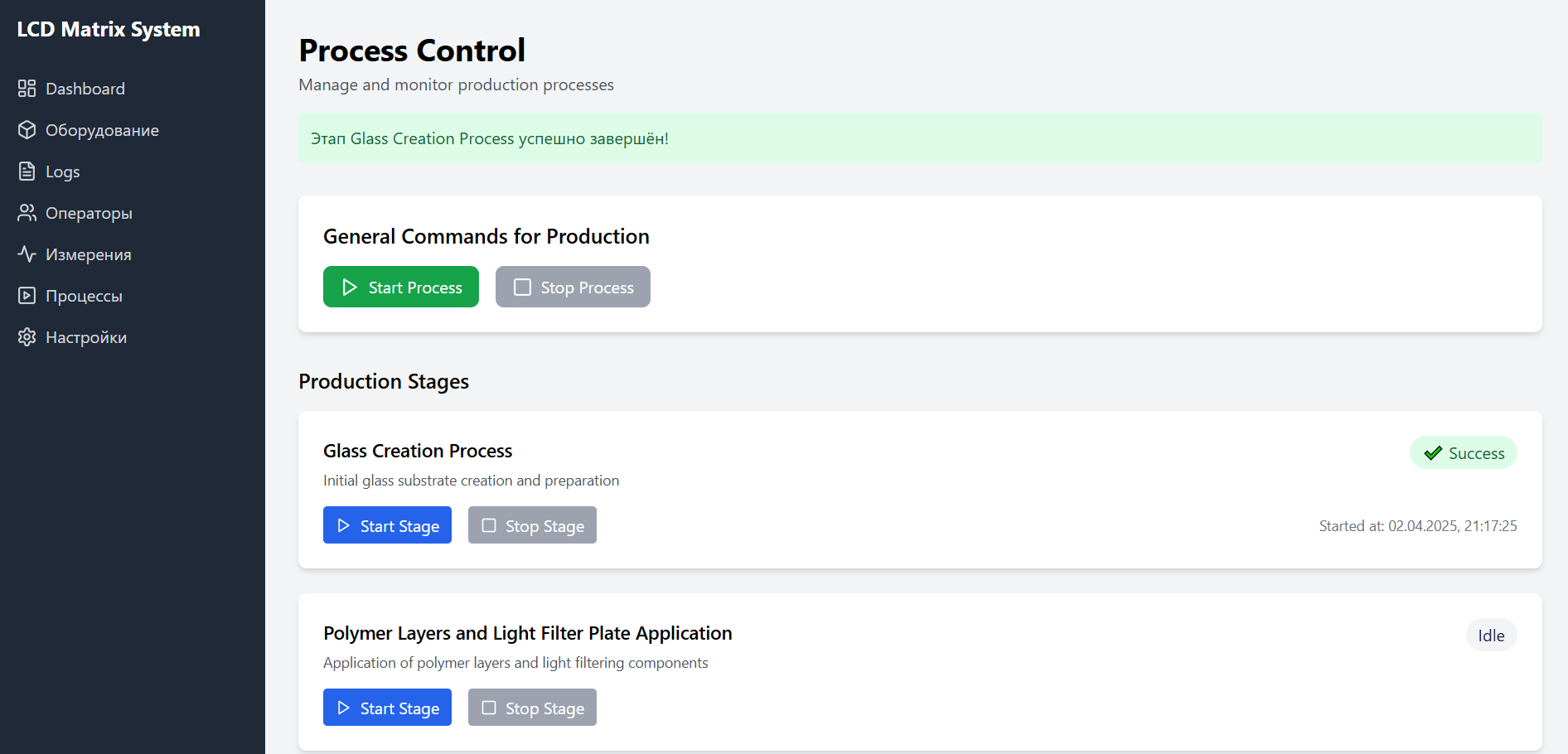
1. Запуск технологических процессов. Необходимо проверить инициирование запуска этапов производства.

Для реализации данной функции было создана страница «Процессы». На ней пользователь может запустить необходимые производственные процессы. Результат проведения тестирования представлен на рисунках 4.3.



**Рисунок 4.3 — Старт процесса производства стекла**

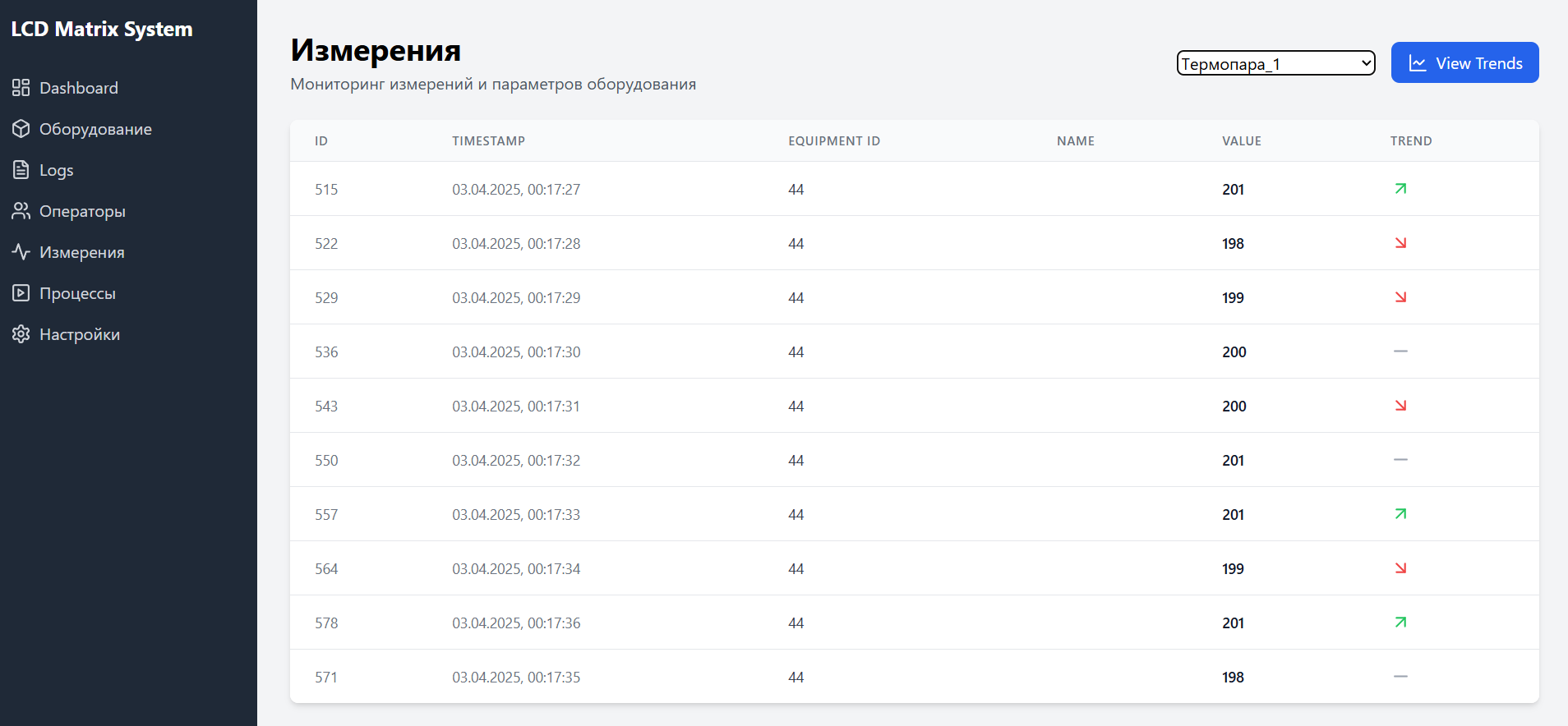
1. Отслеживание технологического процесса. Получение данных с датчиков в режиме реального времени. Отображение результата тестирования сразу же после завершения тестирования, без обновления страницы. Результат выполнения процесса представлен на рисунке 4.4.



**Рисунок 4.4 — Успешное завершение процесса**

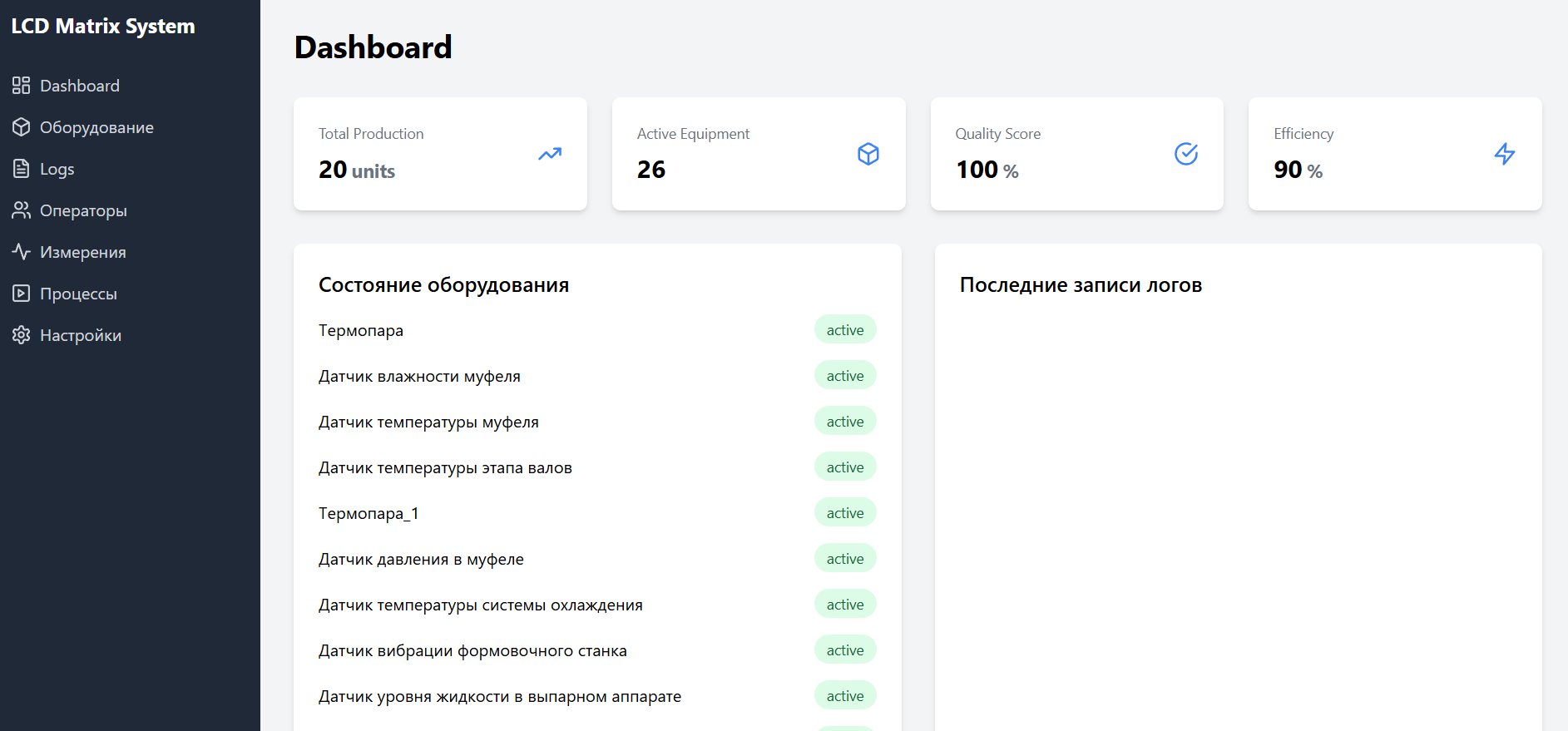
1. Отображение и анализ данных. Необходимо реализовать предоставление информации о параметрах, этапах процесса, выявленных отклонениях.

Для этого была создана страница Измерения с соответствующим эндпоинтом. На странице выводится содержимое таблицы, которая содержит в себе данные об измерении датчиков. Также создан окно с фильтрацией данных по названию устройства. Результат представлен на рисунке 4.5.



**Рисунок 4.5 — Отображение новых записей после проведения процесса создания стекла**

1. Отслеживание результатов производства. Пользователь должен с главной страницы получить всю необходимую информации о создании ЖК-матриц. Результат изменений представлен на рисунке 4.6.

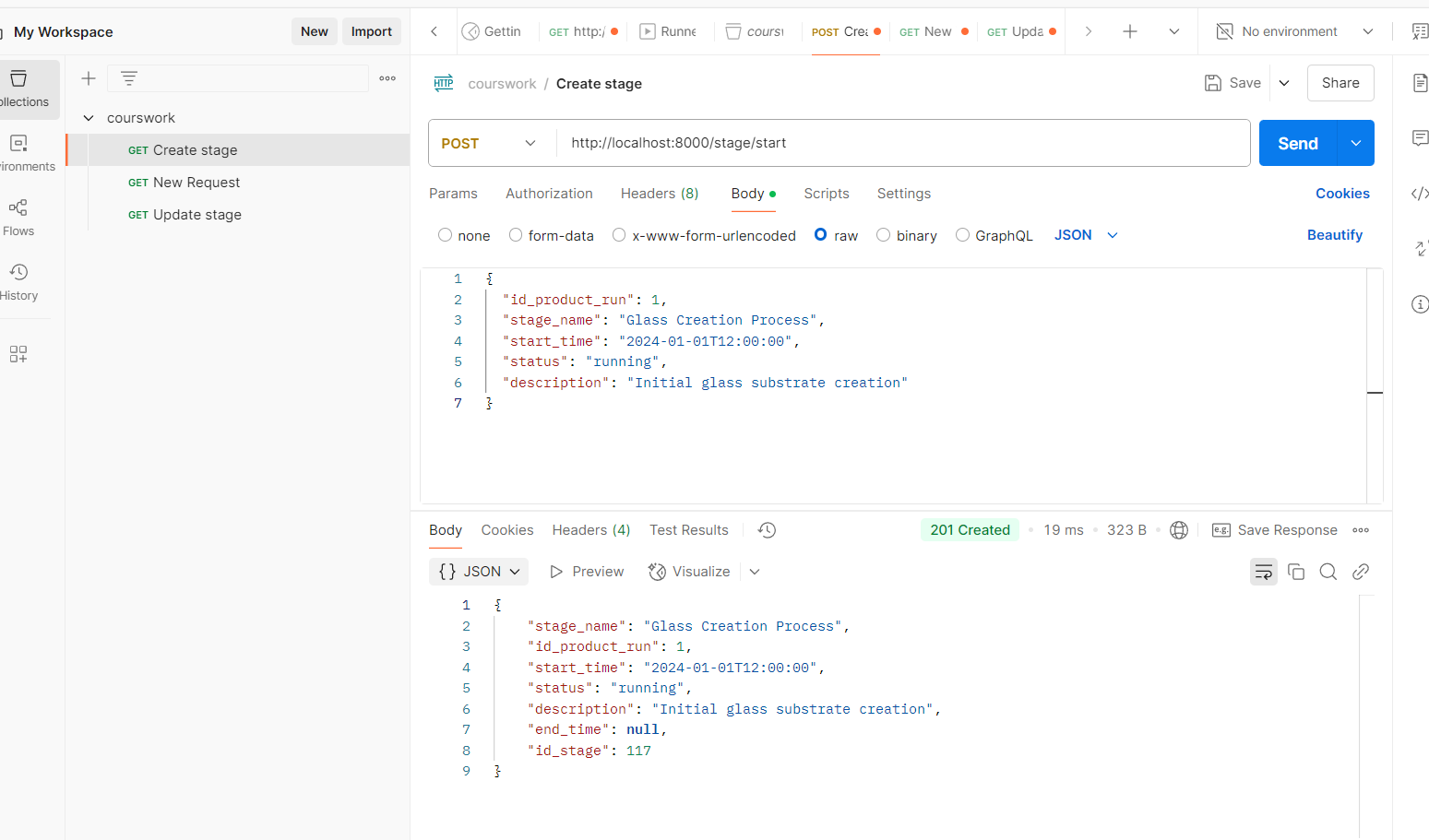


**Рисунок 4.6 — Изменения на главной странице после проведения испытаний**

Также было проведено тестирование на устойчивость системы. Выполнено это было с помощью ПО Postman. С его помощью была создана коллекция из трёх запросов:

* GET-запрос на получения данных об оборудовании
* GET-запрос на получение измерений показаний датчиков
* POST-запрос на запуск нового технологического процесса.

Пример POST-запроса представлен на рисунке 4.7.



**Рисунок 4.7 — Коллекция запросов для нагрузочного тестирования**

Тестирование будет проводиться при следующих настройках: количество итераций — 20, задержка — 0 мс.

Вывод: в результате проведения тестирования было успешно выполнено тестирование функциональности системы согласно разработанным тест-кейсам. Также была выполнена проверка обработки HTTP-запросов с использованием инструмента Postman.

# ЗАКЛЮЧЕНИЕ

В результате работы был рассмотрен технологический процесс создания ЖК-матриц, а также выявлены функциональные требования к автоматизированной системе. Была построена DFD-диаграмма, в которой были выявлены потоки информации, смоделирована логическая схема базы данных, в которой представлены основные таблицы, которые будут функционировать в разрабатываемом ПО, а также архитектура данной разрабатываемой системы. В ходе используемых средств разработки было создано приложение, отвечающее поставленным требованиям. Для достижения реального времени был использован WebSocket из фреймворка FastAPI. В результате проведения тестирования было успешно выполнено тестирование функциональности системы согласно разработанным тест-кейсам. Также была выполнена проверка обработки HTTP-запросов с использованием инструмента Postman.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* 1. IDEF0. Integrated Definition for Function Modeling. — National Institute of Standards and Technology. — 1993 (Официальная документация по нотации IDEF0, использованной для моделирования процессов).
  2. Как производят матрицы для мониторов и телевизоров [Электронный ресурс] // URL: <https://kaprionika.ru/blog/kak-proizvodyat-matritsy-dlya-monitorov-i-televizorov/> (дата обращения: 20.03.2025)
  3. Как производят матрицы для мониторов и телевизоров [Электронный ресурс] // URL: <https://club.dns-shop.ru/blog/t-132-televizoryi/83461-kak-proizvodyat-matritsyi-dlya-monitorov-i-televizorov/> (дата обращения: 20.03.2025)
  4. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. — Addison-Wesley, 2005. — 496 с.
  5. Fowler M. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2002. — 560 с.
  6. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. — Addison-Wesley, 1994. — 416 с.
  7. PostgreSQL Documentation. // URL: https://www.postgresql.org/docs/ (дата обращения: 28.03.2023).
  8. WebSocket API Specification. // URL: https://www.w3.org/TR/websockets/ (дата обращения: 26.03.2023).