Оглавление

[Введение 5](#_Toc532340454)

[1. Техническое задание. Общие положения 6](#_Toc532340455)

[1.1 Полное наименование системы и ее условное обозначение 6](#_Toc532340456)

[1.2 Номер договора (контракта) 6](#_Toc532340457)

[1.3 Наименования организации-заказчика и организаций-участников работ 6](#_Toc532340458)

[1.4 Перечень документов, на основании которых создается система 6](#_Toc532340459)

[1.5 Плановые сроки начала и окончания работы по созданию системы 6](#_Toc532340460)

[1.6 Источники и порядок финансирования работ 6](#_Toc532340461)

[1.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы 7](#_Toc532340462)

[1.8 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ 7](#_Toc532340463)

[1.9 Определения, обозначения и сокращения 7](#_Toc532340464)

[2. Назначение и цели создания системы 8](#_Toc532340465)

[1.10 Назначение системы 8](#_Toc532340466)

[1.11 Цели создания системы 8](#_Toc532340467)

[1.2 Характеристика объекта автоматизации 8](#_Toc532340468)

[1.3 Требования к системе 9](#_Toc532340469)

[1.4 Требования к системе в целом 9](#_Toc532340470)

[1.4.1 Требования к структуре и функционированию системы 9](#_Toc532340471)

[1.4.2 Требования к численности и квалификации персонала системы 9](#_Toc532340472)

[1.4.3 Показатели назначения 10](#_Toc532340473)

[1.4.4 Требования к надежности 10](#_Toc532340474)

[1.4.5 Требования к безопасности 10](#_Toc532340475)

[1.4.6 Требования к эргономике и технической эстетике 10](#_Toc532340476)

[1.4.7 Требования к транспортабельности для подвижных АС 11](#_Toc532340477)

[1.4.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы 11](#_Toc532340478)

[1.4.9 Требования к защите информации от несанкционированного доступа 11](#_Toc532340479)

[1.4.10 Требования по сохранности информации при авариях 11](#_Toc532340480)

[1.4.11 Требования к защите от влияния внешних воздействий 11](#_Toc532340481)

[1.4.12 Требования к патентной чистоте 12](#_Toc532340482)

[1.4.13 Требования по стандартизации и унификации 12](#_Toc532340483)

[1.4.14 Дополнительные требования 12](#_Toc532340484)

[1.5 Требования к функциям (задачам), выполняемым системой 12](#_Toc532340485)

[1.6 Требования к видам обеспечения 12](#_Toc532340486)

[1.6.1 Требования к математическому обеспечению системы 13](#_Toc532340487)

[1.6.2 Требования информационному обеспечению системы 13](#_Toc532340488)

[1.6.3 Требования к лингвистическому обеспечению системы 13](#_Toc532340489)

[1.6.4 Требования к программному обеспечению системы 13](#_Toc532340490)

[1.6.5 Требования к техническому обеспечению 13](#_Toc532340491)

[1.6.6 Требования к метрологическому обеспечению 13](#_Toc532340492)

[1.6.7 Требования к организационному обеспечению 13](#_Toc532340493)

[1.6.8 Требования к методическому обеспечению 14](#_Toc532340494)

[1.4 Состав и содержание работ по созданию (развитию) системы 14](#_Toc532340495)

[1.5 Порядок контроля и приемки системы 15](#_Toc532340496)

[5.1 Виды, состав, объем и методы испытаний системы 15](#_Toc532340497)

[5.2 Общие требования к приемке работ по стадиям 15](#_Toc532340498)

[5.3 Статус приемочной комиссии 15](#_Toc532340499)

[1.6 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие 15](#_Toc532340500)

[1.7 Требования к документированию 16](#_Toc532340501)

[ Техническое задание в соответствии с ГОСТ 34.602-89 16](#_Toc532340502)

[ Расчетно-пояснительную записку, включающую исследовательская часть, проектно-конструкторскую часть и проектно-технологическая часть. Расчетно-пояснительная записка выполняется с учетом требований, предусмотренных ГОСТ 7.32-2001 и 2.105-95. 16](#_Toc532340503)

[2.1 Постановка задачи проектирования 17](#_Toc532340504)

[2.2 Описание предметной области 17](#_Toc532340505)

[2.3 Анализ аналогов. 20](#_Toc532340506)

[2.4 Перечень задач, подлежащих решению в процессе разработки 20](#_Toc532340507)

[3. Проектно-конструкторская часть 22](#_Toc532340508)

[3.1. Разработка структуры приложения. 22](#_Toc532340509)

[3.2. Разработка алгоритмов обработки информации. 24](#_Toc532340510)

[3.3. Логическая схема базы данных 26](#_Toc532340511)

[3.4. Разработка интерфейса взаимодействия пользователя с системой. 28](#_Toc532340512)

[3.5. Разработка архитектуры приложения. 33](#_Toc532340513)

[4. Проектно-техническая часть 35](#_Toc532340514)

[4.1. Проектирование начального и тестового наполнения базы данных. 35](#_Toc532340515)

[4.2. Технологические решения, поддерживающие эксплуатационный цикл программы 35](#_Toc532340516)

[Заключение 36](#_Toc532340517)

[Список использованных источников 37](#_Toc532340518)

[ПРИЛОЖЕНИЕ 38](#_Toc532340519)

# Введение

В ходе выполнения курсовой работы требуется решить поставленные задачи:

* + Приобрести навыки системного анализа и моделирования, ведения научно-исследовательской, проектно-конструкторской и проектно-технологической деятельности.
  + Разработать работоспособное программное приложение, автоматизирующее деятельность тепличного комплекса.
  + Разработать комплект технической документации к проекту

Цель курсовой работы:

Формирование практических навыков по разработке и реализации программного приложения с использованием систем управления базами данных (СУБД).

Приложение является актуальным, так как позволяет средствами объектно-ориентированного программирования, технологии Windows Presentation Foundation, реляционной СУБД PostgreSQL автоматизировать сложную систему работы тепличного комплекса, обеспечить централизованное хранение данных, а также возможность их последующей обработки и анализа.

# Техническое задание. Общие положения

## Полное наименование системы и ее условное обозначение

Система автоматизации тепличного комплекса «Красно Солнышко».

## Номер договора (контракта)

На выполнение курсовой работы договор не заключается.

## Наименования организации-заказчика и организаций-участников работ

Калужский филиал государственного федерального бюджетного образовательного учреждения высшего образования «Московский государственный технический университет имени Н.Э. Баумана» (национальный исследовательский университет).

## Перечень документов, на основании которых создается система

Система создается на основании государственных стандартов оформления научно-исследовательских работ и технической документации, а именно:

* ГОСТ 34.602-89,
* ГОСТ 7.32-2001,
* ГОСТ 2.105-95,
* А также на основании настоящего Технического Задания.

Иных документов не предусмотрено.

## Плановые сроки начала и окончания работы по созданию системы

Срок начала работы над проектом: 05.09.2018.

Срок окончания работы: 12.12.2018.

## Источники и порядок финансирования работ

В процессе выполнения курсовой работы финансирование не производится.

## Порядок оформления и предъявления заказчику результатов работ по созданию системы

В сроки, установленные учебным планом, требуется выполнять следующие этапы по созданию системы:

* 4 неделя (26.09.2018)

Утверждение темы, задания на разработку, технического задания, описание предметной области и требований к системе, анализ существующих аналогов системы, утверждение концептуальной модели обрабатываемых данных и прототипа интерфейса

* 7 неделя (17.10.2018)

Оформление введения и исследовательской части, логической и физической схемы базы данных, оформление и согласование работы макета системы с базой данных

* 10 неделя (07.11.2018)

Оформление проектно-конструкторской документации, демонстрация работающего приложения, оформление презентации о проделанной работе, тестирование и отладка приложения, оформление эксплуатационной документации.

* 14 неделя (01.12.2018)

Финальная отладка программы, презентация готового проекта.

Кроме того, рекомендуется хранить актуальную копию проекта на одном из веб-сервисов хостинга IT-проектов для обеспечения возможности контроля за его выполнением.

## Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ

Во время разработки ТЗ был использован ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы.

## Определения, обозначения и сокращения

В настоящем Техническом Задании используются следующие сокращения:

ТЗ – Техническое Задание,

ТК – Тепличный Комплекс.

# 2. Назначение и цели создания системы

## Назначение системы

Требуется разработать программу по автоматизации тепличного комплекса, состоящего из трех теплиц, снабженных специализированным сельскохозяйственным оборудованием, в т. ч. датчиками температуры и влажности в теплице, а также резервуаром с водой.

## Цели создания системы

Данное программное обеспечение по автоматизации тепличного комплекса «Красно-Солнышко» разрабатывается в целях повышения эффективности производства на данном предприятии через систему автоматизированного контроля ошибок, увеличения сбора урожая за счёт увеличения объемов статистической информации, увеличения эффективности труда служащих через систему задач, автоматического контроля за состоянием резервуаров для воды.

# 1.2 Характеристика объекта автоматизации

Тепличный комплекс представляет из себя три независимые теплицы, в каждой из которых произрастает сельскохозяйственные культуры.

Для построения подобного решения и достижения нужного результата требуются три основных компонента: управление датчиками, которые считывают те или иные параметры; разрабатываемая автоматизированная система, или АС, для сбора и обработки этой информации, а также технологии передачи данных, призванные связать две других составляющих. Перечисленного достаточно для качественного мониторинга состояния воздуха и почвы, регуляции работы автоматики и всех этапов выращивания овощей.

## Требования к системе

## Требования к системе в целом

### Требования к структуре и функционированию системы

#### Перечень подсистем, их назначение и основные характеристики

В систему автоматизации ТК «Красно Солнышко» входят следующие подсистемы:

* система оповещения технического персонала о сбоях в работе теплиц,
* система статистического контроля состояния влажности почвы и воздуха, температуры грунта, температуры воздуха внутри и снаружи теплицы, а также освещенности внутри теплицы,
* система учета расходных материалов (грунта, удобрений, воды в резервуарах)
* система ведения учетных записей технических работников для выполнения ими текущих задач.

#### Требования к способам и средствам связи для информационного обмена между компонентами системы

Персональные компьютеры служащих требуется соединить в локальную сеть, а информацию с датчиков получать с помощью одной из возможных систем беспроводной передачи информации (система LoRa или подобные ей).

### Требования к численности и квалификации персонала системы

К численности персонала требований не предъявляется, однако среди них должны присутствовать:

* техник (работник, устраняющий технические неисправности),
* рабочий теплиц (осуществляющий обработку показаний датчиков и на их основании принимающий решения),
* менеджер по статистике, обрабатывающий информацию о том, как те или иные условия отразились на количестве собранного урожая.

### Показатели назначения

В результате работы требуется получить работоспособную программу, корректно реагирующую на ошибки, допускаемые персоналом во время ее использования, а также на изменения условий ее использования, как то: ввод новой выращиваемой культуры, изменения модели датчиков или способа получения информации с них.

Соответствие системы этим требованиям определяет ее показатели назначения удовлетворительными.

### Требования к надежности

В случае, если любой пользователь системы допускает ошибку при ее использовании, он должен быть уведомлен об этом, причем система должна оставаться работоспособной.

Недопустимы потери хранимой информации вследствие непреодолимых факторов (например, аварийного отключения электроэнергии), кроме физического уничтожения носителей информации.

### Требования к безопасности

Система предназначается для использования в нормальных условиях, воздействие угрожающих целостности системы факторов минимально. Защита датчиков происходит в соответствии с условиями их использования в сопроводительной документации к ним и к разрабатываемой системе прямого отношения не имеет.

### Требования к эргономике и технической эстетике

Пользовательский интерфейс системы для каждого служащего должен быть понятен пользователю, должны отсутствовать надписи, кнопки, поля ввода и прочие элементы интерфейса, которые могут ввести пользователя в заблуждение. Также не рекомендуется использовать элементы пользовательского интерфейса, чье назначение неочевидно или использование которых не имеет результата.

### Требования к транспортабельности для подвижных АС

Разрабатываемая система не является подвижной.

### Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Система предназначена для работы в нормальных условиях, особых требований к эксплуатации не имеет. Должна быть портируема на стационарные компьютеры с операционной системой семейства Windows, обслуживание системы производить по мере необходимости: в случае возникающих фатальных ошибок или таких изменений в процессе производства, что дальнейшее использование программы будет неэффективно или невозможно.

### Требования к защите информации от несанкционированного доступа

Предприятие не имеет уровня секретности информации, хранимые данные не обладают ни одним из статусов защищаемых государством сведений. Поэтому необходимости в строгих мерах по защите информации от несанкционированного доступа нет.

### Требования по сохранности информации при авариях

В случае аварийного отключения электропитания или иных аварийных ситуациях, связанных с возможностью потери информации, в системе должны сохраниться данные полученные не ранее дня, предшествующего произошедшим событиям (кроме случаев физического уничтожения носителей информации).

### Требования к защите от влияния внешних воздействий

От внешних воздействий защищать систему не требуется.

### Требования к патентной чистоте

При разработке системы интересы и авторские права других стран не должны быть затронуты.

### Требования по стандартизации и унификации

Система должна быть используема на персональных компьютерах с операционной системой Windows XP или выше. Сопроводительные документы должны быть оформлены в соответствии с вышеперечисленными государственными стандартами оформления технической документации.

### Дополнительные требования

Система должна сопровождаться всей необходимой документацией и руководством пользователя.

## Требования к функциям (задачам), выполняемым системой

Программа должна обеспечивать возможность выполнения следующих функций:

* контроль показаний датчиков,
* ведение статистики о закупке и продаже семян, реализации готовой продукции, а также температурных условиях и влажности во время выращивания овощей,
* ведение учетных записей пользователей, отражающих их обязанности,
* система оповещения о сбоях и ошибках,
* контроль количества воды в емкостях для полива, а также овощей, готовых к реализации.

## Требования к видам обеспечения

Система должна быть совместима с Windows XP и выше, техническое оснащение персональных компьютеров должно соответствовать требованиям этой операционной системы.

Модели датчиков, необходимых для работы, выбираются предприятием и для разрабатываемой системы не должны иметь большого значения, кроме спектра информации, которую они могу предоставить.

### Требования к математическому обеспечению системы

Требования к математическому обеспечению системы отсутствуют.

### Требования информационному обеспечению системы

Хранение и обслуживание данных в системе должно происходить с помощью одной из существующих систем управления базами данных. Информация должна сохранять целостность, не должно возникать аномалий при попытке удаления, изменения и добавления данных.

### Требования к лингвистическому обеспечению системы

Программа должна быть написана на одном из высокоуровневых языков программирования. Приоритетным является использование языка C#. Оформление пользовательского интерфейса производить про помощи системы Windows Presentation Foundation, а данные организовать в реляционную базу данных.

### Требования к программному обеспечению системы

Иных требований к программному обеспечению, кроме освещенных в текущем ТЗ, система не имеет.

### Требования к техническому обеспечению

Иных требований к техническому обеспечению, кроме освещенных в текущем ТЗ, система не имеет.

### Требования к метрологическому обеспечению

С требования к метрологическим характеристикам используемых в системе датчиков ознакомиться в сопроводительной документации к ним. Система должна принимать показания датчиков в пределах допустимой погрешности.

### Требования к организационному обеспечению

Использующие систему подразделения должны выполнять свои рабочие обязанности в пределах своей сферы ответственности. В случае пользовательской ошибки система должна предупредить об этом пользователя, не теряя при этом работоспособности. Взаимодействие персонала с системой (и между собой в рамках системы) должно осуществляться в рабочем порядке.

### Требования к методическому обеспечению

Разработку системы требуется сопровождать расчетно-пояснительной запиской и графической частью. Расчетно-пояснительная записка состоит из сле­дующих частей:

1. Техническое задание.
2. Исследовательская часть. (Научно-исследовательская часть.)
3. Проектно - конструкторская часть.
4. Проектно-технологическая часть

Применяемые стандарты:

* ГОСТ 34.602-89,
* ГОСТ 7.32-2001,
* ГОСТ 2.105-95.

# 1.4 Состав и содержание работ по созданию (развитию) системы

1. Создание прототипа интерфейса. Экспертиза: согласование прототипа с заказчиком в письменном виде. (Срок 4 неделя)
2. Создание концептуальной модели хранимой информации. Экспертиза: согласование прототипа с заказчиком в письменном виде. (Срок 7 неделя)
3. Разработка физической и логической схемы хранимой информации. Экспертиза: демонстрация проделанной работы заказчику. (Срок 7 неделя)
4. Разработка макета программы с базой данных. Экспертиза: демонстрация проделанной работы заказчику. (Срок 10 неделя)
5. Отладка и устранение ошибок программы. Экспертиза: демонстрация проделанной работы заказчику. (Срок 14 неделя)

# 1.5 Порядок контроля и приемки системы

В сроки, предусмотренные учебным планом, производится контроль состояния разработки системы, разбираются ошибки и производится корректировка текущей работы.

По окончанию работы следует представить отчет о проделанной работе в виде расчетно-пояснительной записки, графической презентации работы, а также непосредственно работающего приложения.

## 5.1 Виды, состав, объем и методы испытаний системы

В случае необходимости применить одну или несколько методик тестирования для демонстрации устойчивости программы.

## 5.2 Общие требования к приемке работ по стадиям

Прием работы осуществляется в сроки, оговоренные в настоящем ТЗ.

## 5.3 Статус приемочной комиссии

Прием работы на всех этапах осуществляется непосредственным руководителем курсовой работы. По окончании разработки работоспособное приложение демонстрируется на комиссии.

# 1.6 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

В момент установки системы в ТК в нем уже должны присутствовать работоспособные датчики и персональные компьютеры пользователей. Если это условие не соблюдается, необходимо провести работы по подготовке объекта автоматизации к вводу системы в действие. Иных работ не предусматривается.

# Требования к документированию

Требуется предоставить:

## Техническое задание в соответствии с ГОСТ 34.602-89

## Расчетно-пояснительную записку, включающую исследовательская часть, проектно-конструкторскую часть и проектно-технологическая часть. Расчетно-пояснительная записка выполняется с учетом требований, предусмотренных ГОСТ 7.32-2001 и 2.105-95.

**2.** Исследовательская часть

1. Постановка задачи проектирования **[1, 3]**

Основной задачей проектирования является создание работоспособного программного приложения на языке C# с помощью системы управления базами данных PostgresSQL.

1. Описание предметной области [7]

В автоматизируемом ТК на территории трех теплиц выращивается перечень сельскохозяйственных культур (огурцов и томатов), каждой из которых на определенной стадии развития необходимо обеспечить необходимые для роста, развития или плодоношения условия. Эти условия должны также быть приближенными к оптимальным для данной культуры с целью повышения урожайности, а следовательно, экономической эффективности проекта.

Также можно отметить, что тепличный комплекс – система, инертная во времени, с предсказуемым перечнем регулярно возникающих задач, составление и отслеживание которых тоже подлежит автоматизации.

В процессе анализа предметной области были выделены следующие сущности:

* Теплица. Сущность, хранящая информацию о работниках, закрепленных за этой теплицей и произрастающих в ней культурах.
* Культура. Содержит информацию о виде, сорте, а также список технологических карт, определяющих оптимальные показатели, подходящие этой культуре.
* Технологическая карта. Унифицированная сущность с полями – Название параметра (содержит идентификатор параметра – например, температура воздуха), Стадия развития (описывает, на какой стадии развития эта технологическая карта применима), Оптимальное значение (наиболее предпочтительное значение параметра), Допустимое отклонение (интервал значений, выход за пределы которого нежелателен, но не повлечет за собой тяжелых последствий), Предельное отклонение (интервал значений, выход за пределы которого может быть губителен для растения)
* Работник. Содержит информацию о работнике теплицы, его персональных данных, должности, а также данные для входа в систему.
* Задача. Сущность, воплощающая задание, выполняемое работником. Может быть регулярной и храниться в системе, а может задаваться вручную, как, например, отчет об ошибке в технический отдел. Содержит поля с непосредственным текстом задачи, датой ее актуализации, а также отдел, исполняющий данную задачу.
* Растение. Сущность выращиваемой в теплице единицы культуры. Хранит информацию о виде и сорте растения, текущей стадии его развития (0 – рассада, 1 – рост, 2 – цветение, 3 - плодоношение), теплице, в которой растение произрастает и в каком количестве.
* Сорт. Хранит информацию о названии сорта.
* Журнал ошибок. Сущность, обеспечивающая статистику в тех случаях, когда состояние какого-либо параметра вышло за пределы рекомендованного либо критического значения, содержит дату и время возникновения проблемы, а также параметр, отклонившийся от нормы.
* Урожай. Хранит количество собранных овощей с определенного сорта в определенную дату.

Таким образом, в теплице произрастает некоторое количество растений разных сортов, находящихся на разных стадиях своего развития, каждому из которых необходимо обеспечить оптимальные условия выращивания.

Система с помощью датчиков отслеживает состояние параметров в теплице и в случае неблагоприятной ситуации оповещает пользователя о проблеме. Сделать вывод о том, являются ли текущие показатели оптимальными, система может с помощью механизма технологических карт. Если пользователь не в силах самостоятельно урегулировать ситуацию, он может обратиться за помощью в технический отдел с помощью механизма задач. Также пользователь способен контролировать стадии роста и развития растений, высаженных в теплице.

С помощью Журнала ошибок пользователь может проанализировать текущие тенденции выращивания той или иной культуры: оказались ли наиболее оптимальными условия в течении определенного промежутка времени.

Можно выделить типы пользователей:

* Работник теплицы. В его рабочей области будут отражены показатели датчиков, сообщения об отклонениях, если таковые происходят, текущие задачи и перечень Растений, выращиваемых в теплице, за которой закреплен работник. Также присутствует механизм изменения стадии развития растения работником и учет собранного урожая.
* Администратор. В его рабочей области находится перечень ошибок за определенный период, статистика о собранном урожае, а также формы формирования новых задач, ввода новых культур для выращивания и добавления технологических карт.

Помимо всего прочего, в работе ТК присутствует склад, в котором хранится необходимый инвентарь, семена и удобрения, система водоснабжения, отопительная система, отдел бухгалтерии, ведущий учет и реализации готовой продукции, а также уровень заработной платы работников, отдел кадров, отвечающий за наем и увольнение сотрудников, а также смену их должностных обязанностей. Работа всех этих отделов в рамках данного проекта автоматизации не подлежит. Подлежат автоматизации: механизм регулярных задач для повышения эффективности и упрощения взаимодействия между различными отделами комплекса, система контроля условий выращивания культур для увеличения урожайности и получения большей выгоды, а также учет и анализ негативных ситуации и их возможных последствий.

# Анализ аналогов

В ходе изучения предметной области полноценных аналогов системы найдено не было. Однако можно обнаружить системы, выполняющие поставленную задачу частично. Так, например:

* система полуавтоматического капельного полива «Жук», позволяющая через определенные промежутки времени обеспечивать растения водой, а следовательно, поддерживать уровень влажности,
* система автоматического обогрева теплиц Heatline – Грунт, с помощью которой пользователь может не допустить падения температуры воздуха или почвы ниже критической отметки,
* компьютерная программа агрохимик, позволяющая вычислять требующиеся удобрения и дозу их внесения.

* 1. Перечень задач, подлежащих решению в процессе разработки**[4]**

В процессе разработки системы и отладки входящих в неё программ, включая проведение необходимых исследований при создании программного продукта, требуется решить следующие задачи:

1. Провести анализ условий выращивания перечня культур для составления начальных технологических карт,
2. Провести анализ регулярных мероприятий ТК для введения механизма задач,
3. Спроектировать интерфейс пользователей в зависимости от исполняемых ими обязанностей,
4. Организовать структуру хранения данных и реализовать тестовое наполнение базы данных,
5. Реализовать логику приложения,
6. Произвести отладку приложения и представить работоспособную программу.

Таким образом, в результате изучения предметной области и анализа процессов, которые должны происходить в системе, была создана концептуальная модель данных, произведен анализ аналогов системы, а также перечень задач, подлежащих решению в процессе разработки.

# Проектно-конструкторская часть

# Разработка структуры приложения [5]

Приложение будет состоять из двух форм основного функционала: формы работника теплицы и формы администратора системы, а также дополнительных форм для ввода новой информации в систему. Работник теплицы может осуществлять контроль над текущими параметрами теплицы, такими как: температура воздуха, освещенность, влажность воздуха и влажность почвы. Помимо этого, ему доступен перечень растений, культивируемых в его теплице месте с их стадиями развития, а также технологические карты каждого растения, если таковые имеются. Взаимодействуя со списком растений, работник может не только изменить стадию развития того или иного растения, но и представить системе отчет о собранном урожае. Более того, работник имеет возможность добавлять в систему информацию о новых высаживаемых растениях. Все эти процессы выполняются с помощью взаимодействия со служебными формами, результат записывается в базу данных для анализа производительности теплицы.

Помимо отображения текущих и желаемых параметров окружающей среды на форме, система будет в автоматическом режиме формировать отчет об ошибке, если какой-либо из параметров выходит за пределы допустимых значений для того или иного растения. Хранение отчетов о таких ошибках реализовано в базе данных с целью последующей обработки и формирования статистических данных о корреляции условий выращивания и собираемом урожае, а также о нормальном функционировании систем теплицы в целом.

В процессе взаимодействия с формой, работник теплицы может ознакомиться с текущими задачами по поддержанию стабильности работы теплицы. Среди таких задач могут присутствовать как регулярные, хранящиеся в системе, так и назначенные конкретно для каждой из теплиц в случае возникновения непредвиденных ситуации или изменений в работе системы. Работник может отметить задачу как выполненную с занесением результата работ в базу данных для последующего ознакомления администратора.

Второй основной функционирующий компонент системы – это форма администратора комплекса, где он может ознакомиться с количеством собранного урожая за отчетный период, а также с перечнем происходивших в системе ошибок. Взаимодействие с этой формой позволит ему также обновлять, изменять и формировать новые задачи, что служит поддерживанию долговечности и целостности системы во времени. Это осуществляется с помощью второстепенных форм для ввода в систему новой информации. Также администратор способен добавлять в систему информацию о новых культурах для выращивания, а также записывать для них технологические карты.

Таким образом, основными потребителями информации, поступающей из базы данных, являются работники теплиц и администраторы комплекса. Они воспринимают информацию о выращиваемых растениях, текущих задачах, урожайности и произошедших ошибках, и в то же время являются источником информации для базы данных, так как владеют наиболее актуальными данными.

Система разработана в среде программирования MS Visual Studio 2015, в качестве системы управления базами данных используется PostgreSQL. Примерная последовательность разработки системы:

* Формирование физической схемы данных и создание надлежащей структуры базы данных;
* Реализация тестового наполнения базы данных;
* Разработка прототипа интерфейса формы работника теплицы;
* Реализация подключения интерфейса к СУБД с помощью введения уровня бизнес-логики приложения по паттерну MVVM;
* Разработка алгоритмов чтения данных из СУБД;
* Внесение необходимых правок, разработка дополнительных форм для ввода информации;
* Реализация механизма входа в приложение для различных типов работников;
* Разработка формы администратора и всех вспомогательных форм, обеспечивающих ее работу.
* Тестирование и отладка приложения.

# Разработка алгоритмов обработки информации [2, 5]

В систему входит алгоритм проверки текущего состояния параметра среды (рисунок 1) и его соответствия параметрам произрастающих в теплице растений. Хранение информации о необходимых параметрах реализовано в виде технологических карт, каждая из которых хранит собственно название параметра, его оптимальное значение, допустимые отклонения от оптимального значения, при соблюдении которых условия выращивания близки к идеальным, а также предельные отклонения, выход за границы которых грозит тяжелыми последствиями.

Во время поступления информации с датчиков значения контролируемых параметров изменяются, что влечет за собой вызов события об изменении значения соответствующего свойства зависимости. В методе, подписанном на это событие, новое значение параметра проверяется на соответствие оптимальным параметрам всех растений в теплице. Результат проверки записывается в базу данных. При этом необходимо помнить о том, что большинство показателей (исключая температуру), вычисляются в процентах. Поэтому необходимо следить, чтобы не возникало ситуации, когда значение параметра выходит за пределы 100%, что говорит прежде всего о неисправности датчика.

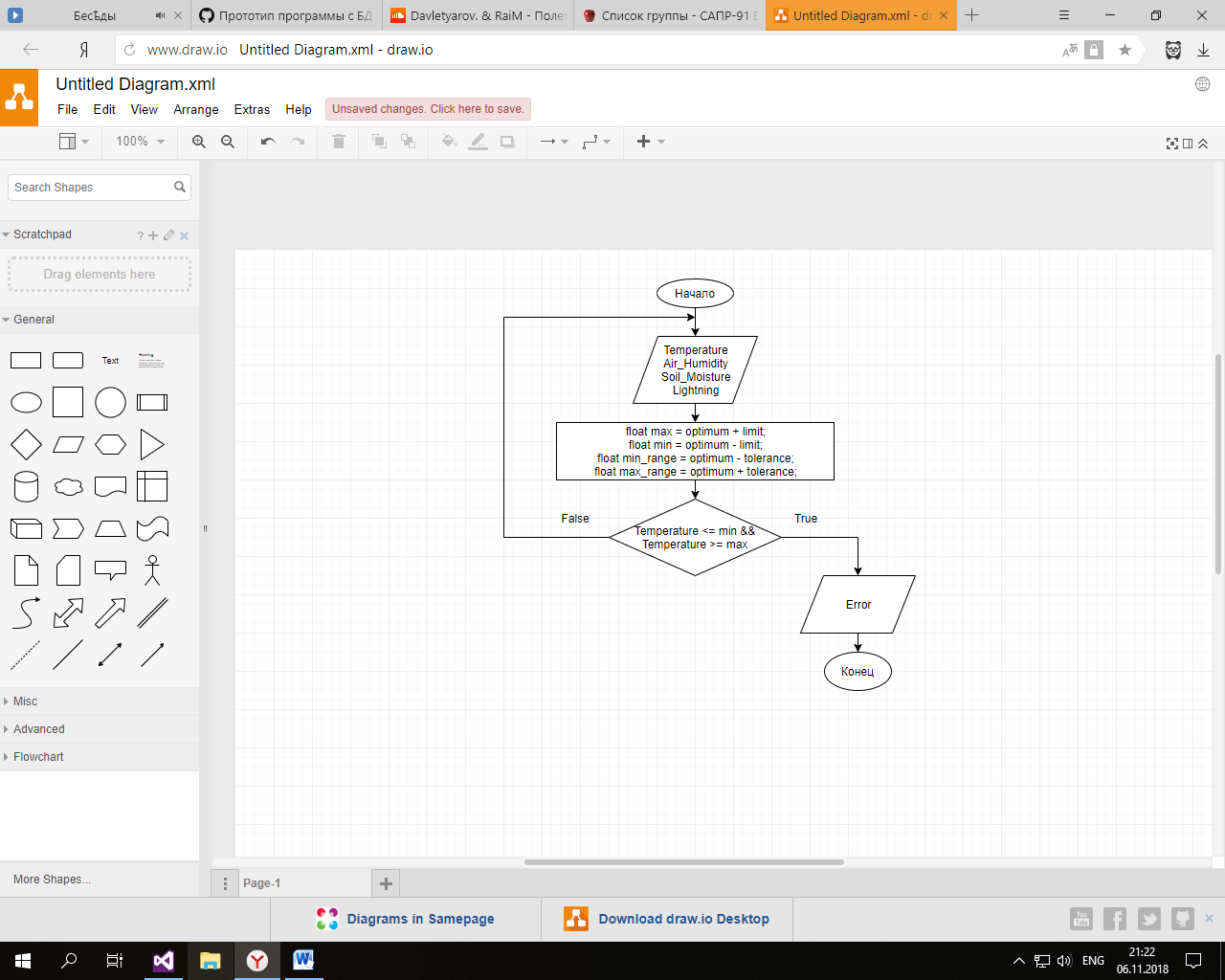


Рисунок 1 – Графическое преставление алгоритма обработки информации, поступающей с датчиков

# Логическая схема базы данных [3, 6]

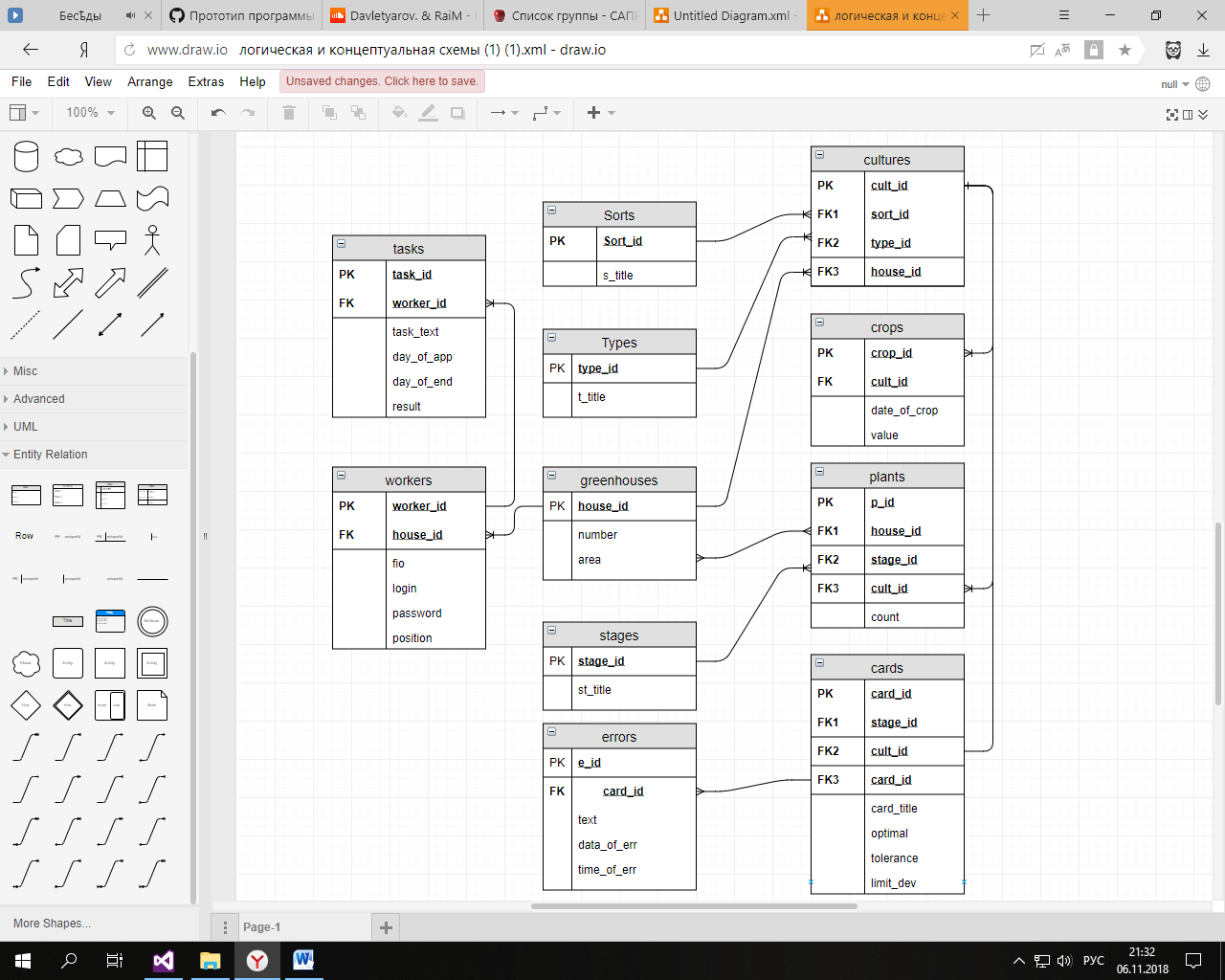


Рисунок 2 – Логическая схема базы данных

Логическая схема проектируемой системы состоит из 12 сущностей, среди которых:

* Sorts – сущность-справочник, хранящий значения сортов выращиваемых культур;
* Types – сущность-справочник, хранящий значения видов выращиваемых культур;
* Stages – сущность-справочник, хранящий перечень стадий, которые проходит растение в ходе своего выращивания;
* Greenhouses – сущность, хранящая информацию о теплицах, входящих в комплекс, их номер и площадь;
* Cultures – сущность, объединяющая значения вышеописанных справочников, чтобы получить информацию о культивируемой культуре и теплицах, в которых она выращивается;
* Plants – сущность, хранящая информацию о произрастающих в данный момент в теплице растений. Удаление отмерших растений из базы данных не происходит, на уровне интерфейса пользователя таким растениям присваивается максимальное значение стадии развития, после чего растение на форме не показывается;
* Cards – сущность, хранящая технологические карты для растений, включает название параметра, его оптимальное значение, допуск и предельное отклонение;
* Errors – сущность, хранящая отчеты о возникающих в системе ошибках при отклонении параметров от их оптимальных значений. Включает дату и время возникновения ошибки, текст, представляющий собой автоматически генерируемую строку на основе той технологической карты, несоответствие которой и было причиной возникновения ошибки;
* Tasks – сущность, хранящая задачи для работников. Включает информацию о дате появления задачи, непосредственно содержание задачи, дату выполнения задачи, а также результат выполнения задачи;
* Workers – сущность, хранящая информацию о работниках теплиц, их персональных данных, занимаемой должности, а также логине и пароля для входа в систему;
* Crops – сущность, хранящая информацию о событии сбора урожая конкретной культуры. Включает поля даты сбора урожая, а также его количество.

Таким образом, логическая схема базы данных показывает, что описываемое ей отношения находятся в третьей нормальной форме, так как ни один ключевой атрибут не находится в транзитивной функциональной зависимости от потенциального ключа.

# Разработка интерфейса взаимодействия пользователя с системой [2, 5]

Приложение состоит из двух форм, реализующих основной функционал, и пяти вспомогательных форм, с помощью которых осуществляется частное взаимодействие с базой данных. Все перечисленные формы представлены на рисунках ниже.

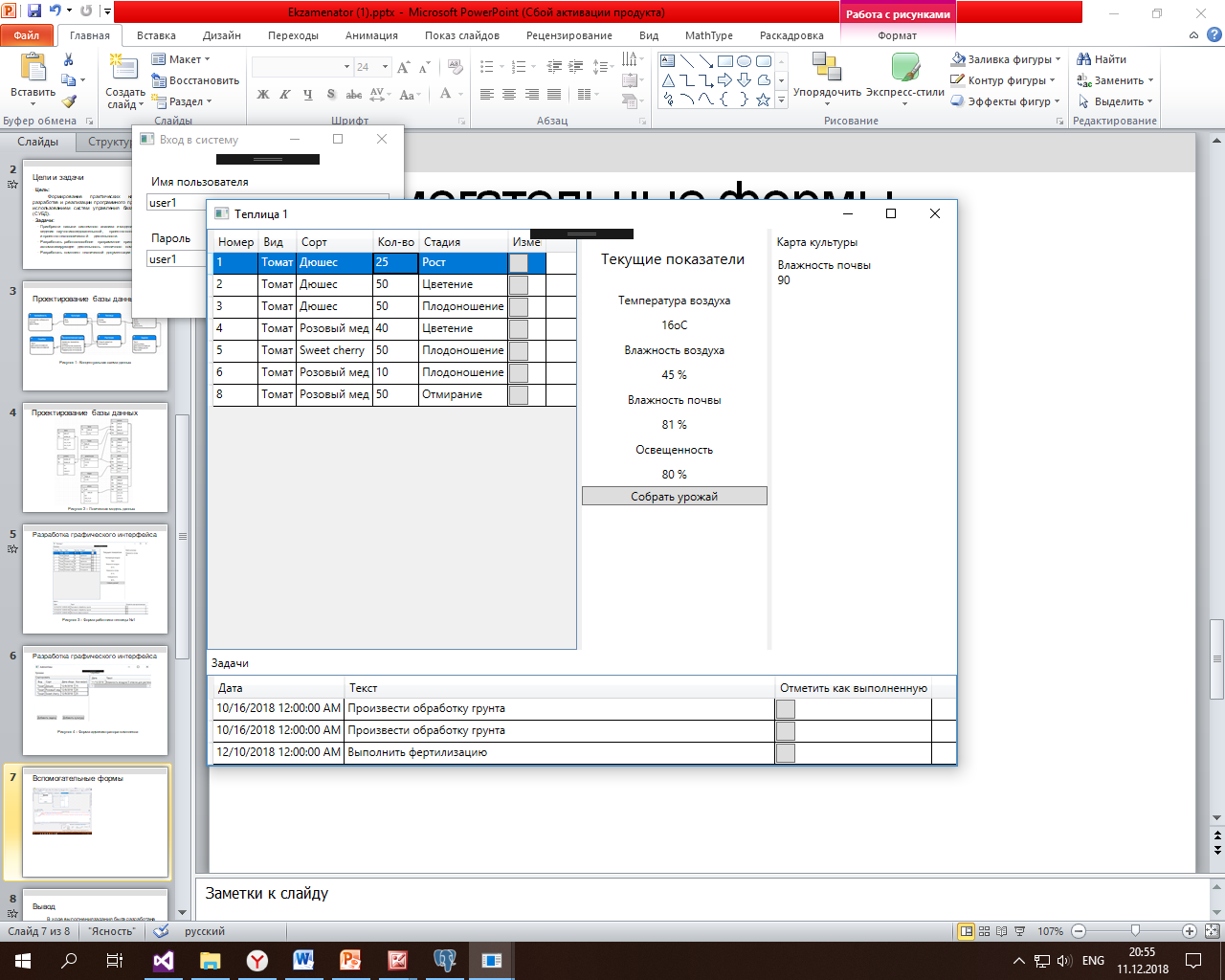


Рисунок 3 – Форма работника теплицы

На рисунке 3 представлен интерфейс формы работника теплицы №1, заполненной тестовыми данными. Форма разделена на 4 функциональные части: слева расположена таблица, демонстрирующая все произрастающие на данный момент культуры в теплице и на какой они стадии развития. В центре располагаются текущие параметры теплицы, получаемые с датчиков, а так же кнопка, с помощью которой пользователь сможет внести в базу данных запись о новом урожае. Справа расположен список технологических карт. При выделении определенного растения в таблице слева в этой области появляется список технологических карт, соответствующий виду, сорту, а так же стадии развития этого растения. На форму выводится только оптимальный показатель.

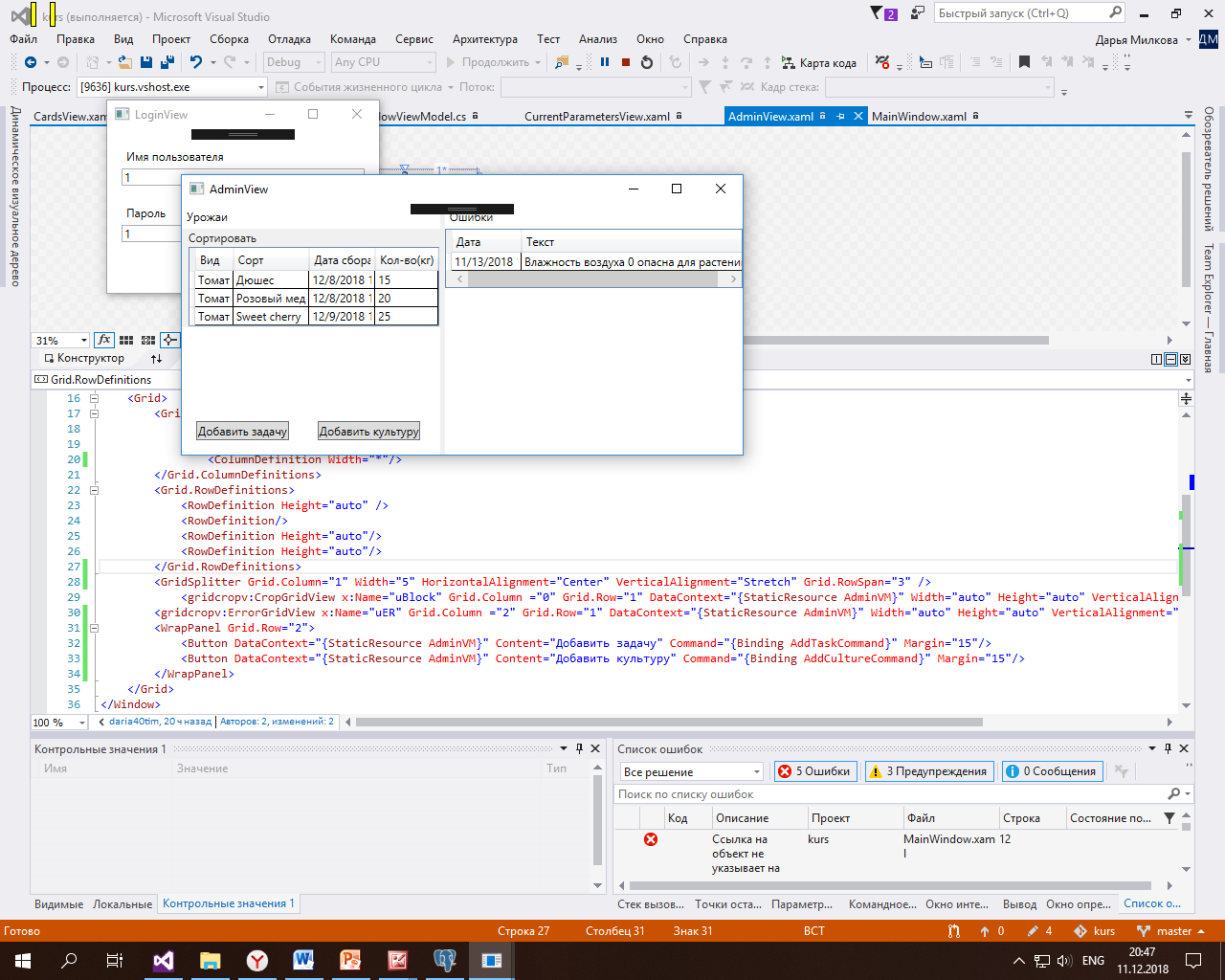


Рисунок 4 – форма администратора комплекса

На рисунке 4 представлена форма администратора тепличного комплекса. Она разбита на две функциональные части. На форме слева представлена таблица всех собранных урожаев, которую пользователь может отфильтровать по дате и количеству собранного урожая. На форме справа расположена таблица ошибок, происходивших в системе для последующего анализа и принятия мер по их устранению. Присутствуют две кнопки, с помощью которых пользователь может вызвать форму добавления задачи (рисунок 5) и форму добавления культуры(рисунок 6) в систему.

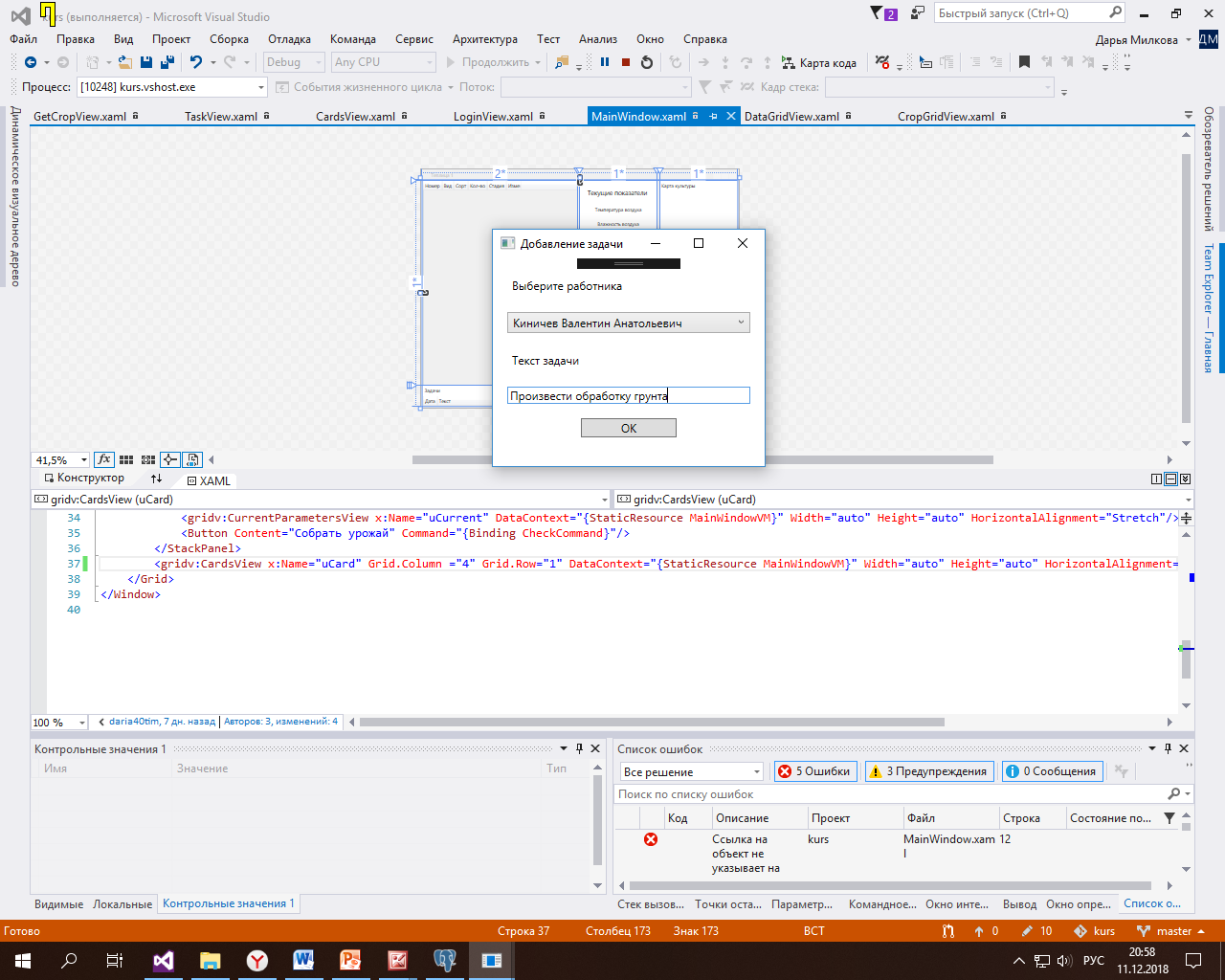


Рисунок 5 – Форма добавления задачи

На форме, представленной на рисунке 5, пользователь может выбрать, кому назначить выполнение задачи из перечня работников, уже зарегистрированных в системе, а так же вписать непосредственно текст задачи.

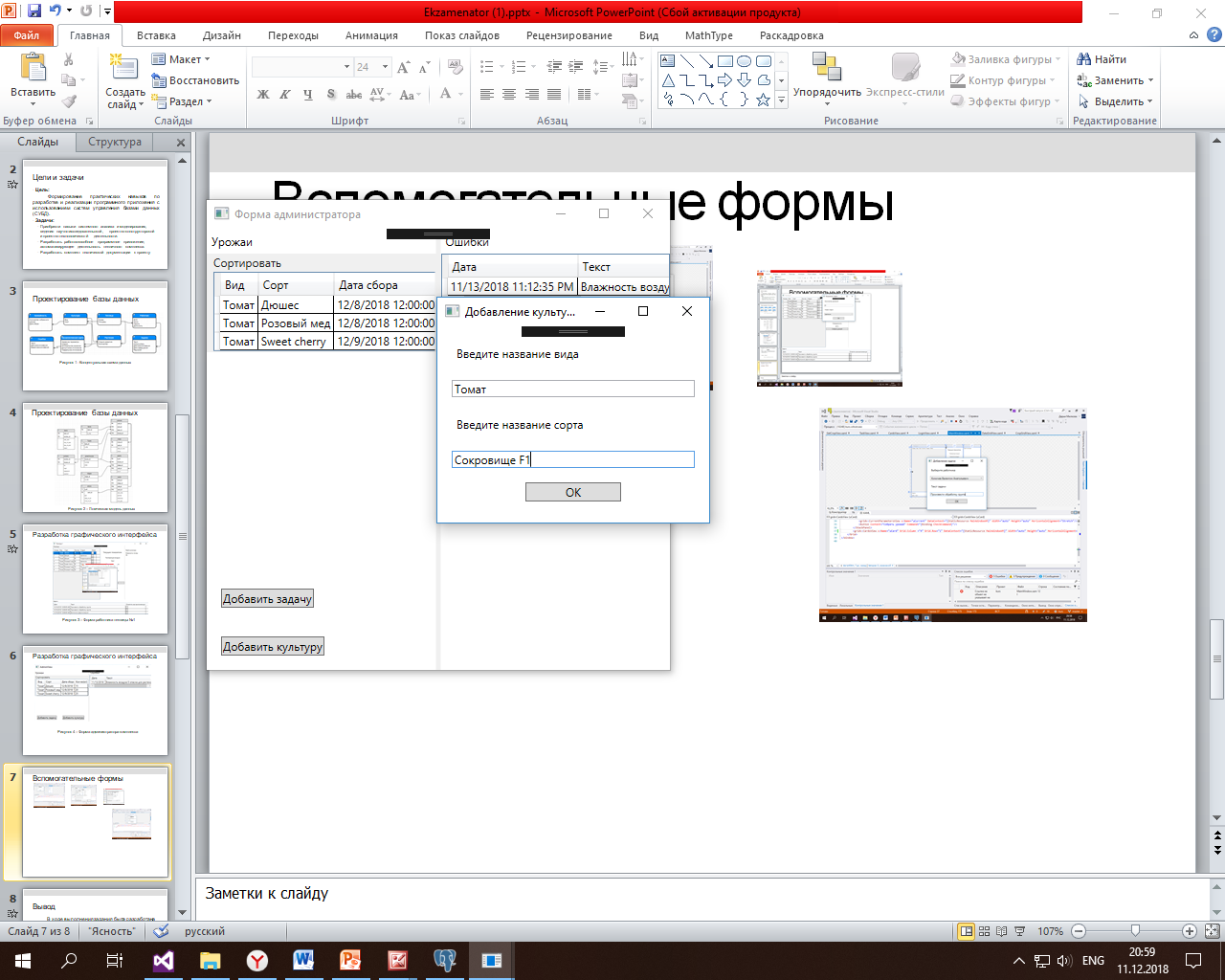


Рисунок 6 – Форма добавления культуры

На форме, представленной на рисунке 6, пользователь может ввести сорт и вид новой культуры, которую необходимо добавить в систему для дальнейшего выращивания и таким образом создать новую запись в таблицах культур, сортов и видов.

Так же вспомогательной формой является форма входа пользователя в систему, приведенная на рисунке 7.

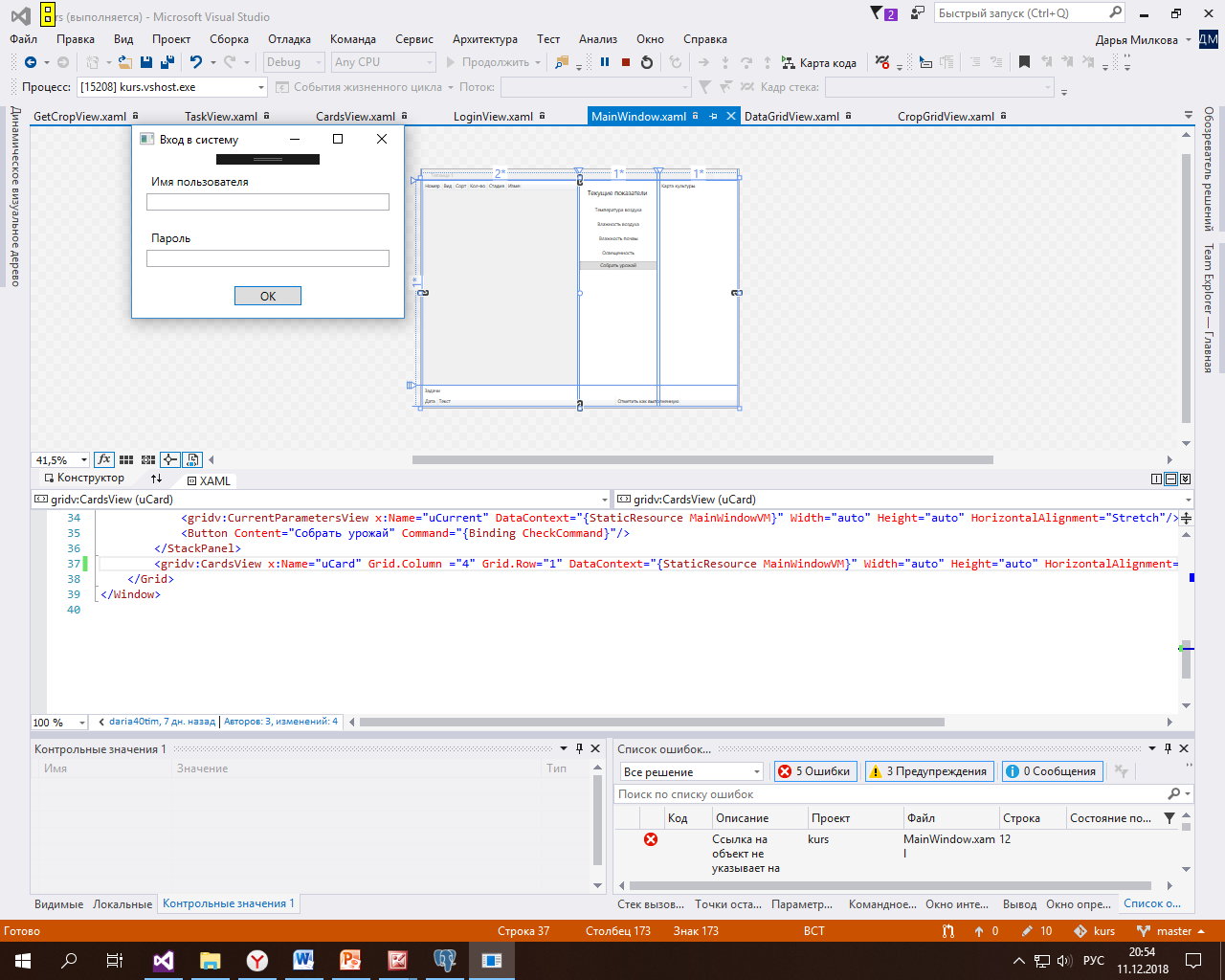


Рисунок 7 – Форма входа в систему

Из формы работника теплицы пользователь может вызвать еще две вспомогательные формы. Одна из них представлена на рисунке 8 – форма изменения стадии растения в теплице по мере его роста и развития.

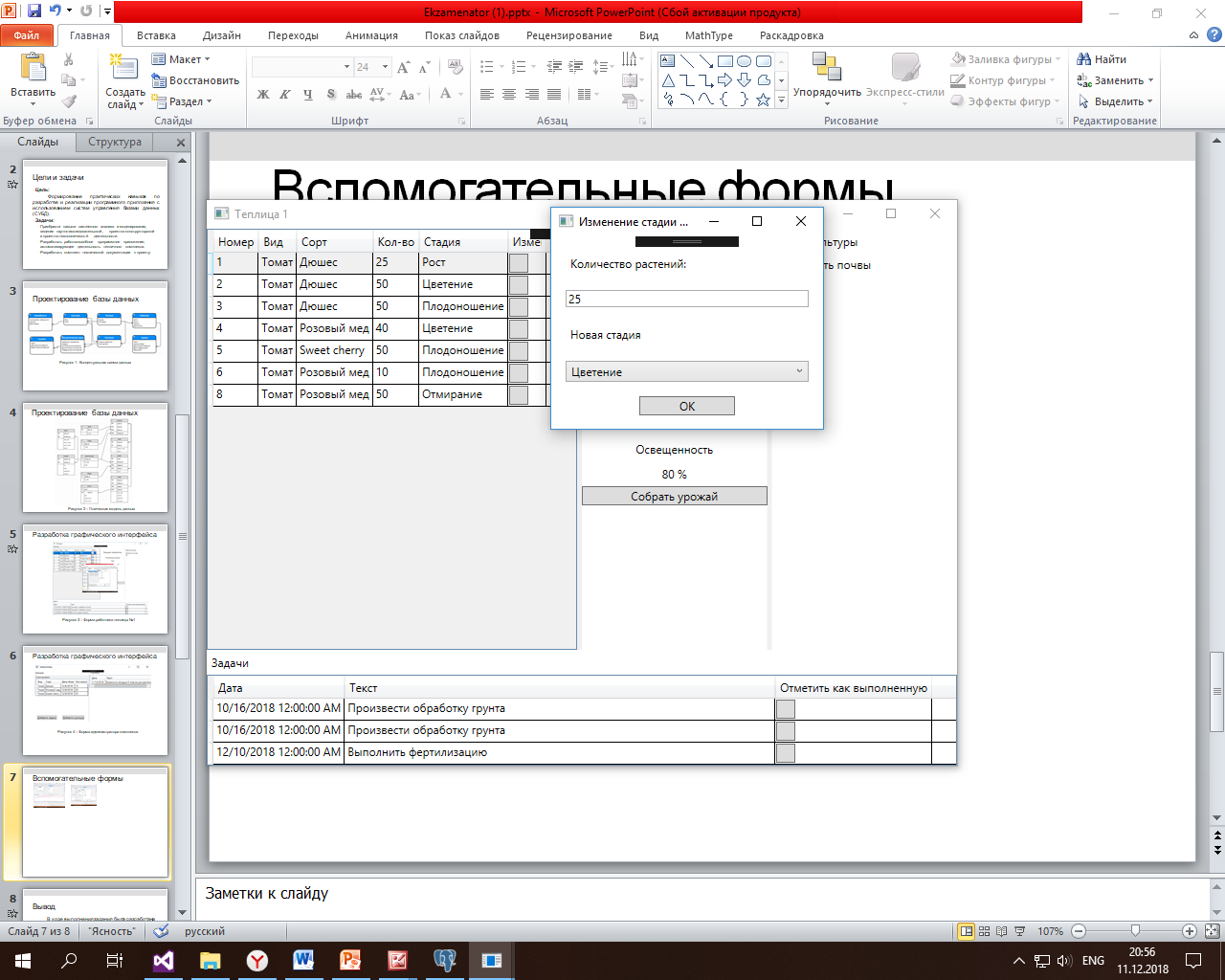


Рисунок 8 – Форма изменения стадии растения

В таблице актуальных растений на каждой записи присутствует кнопка изменения стадии, с помощью которой работник теплицы может легко вносить изменения в состояние теплицы. По нажатии на кнопку в одной из записей будет вызвана форма из рисунка 8, где работник вводит количество растений, перешедших на новую стадию, а также выбирает саму стадию из списка возможных.

Следующая вспомогательная форма представлена на рисунке 9 – форма отчета по собранному урожаю. На этой форме пользователь может выбрать культуру из списка плодоносящих растений его теплицы и ввести информацию о количестве собранного урожая.

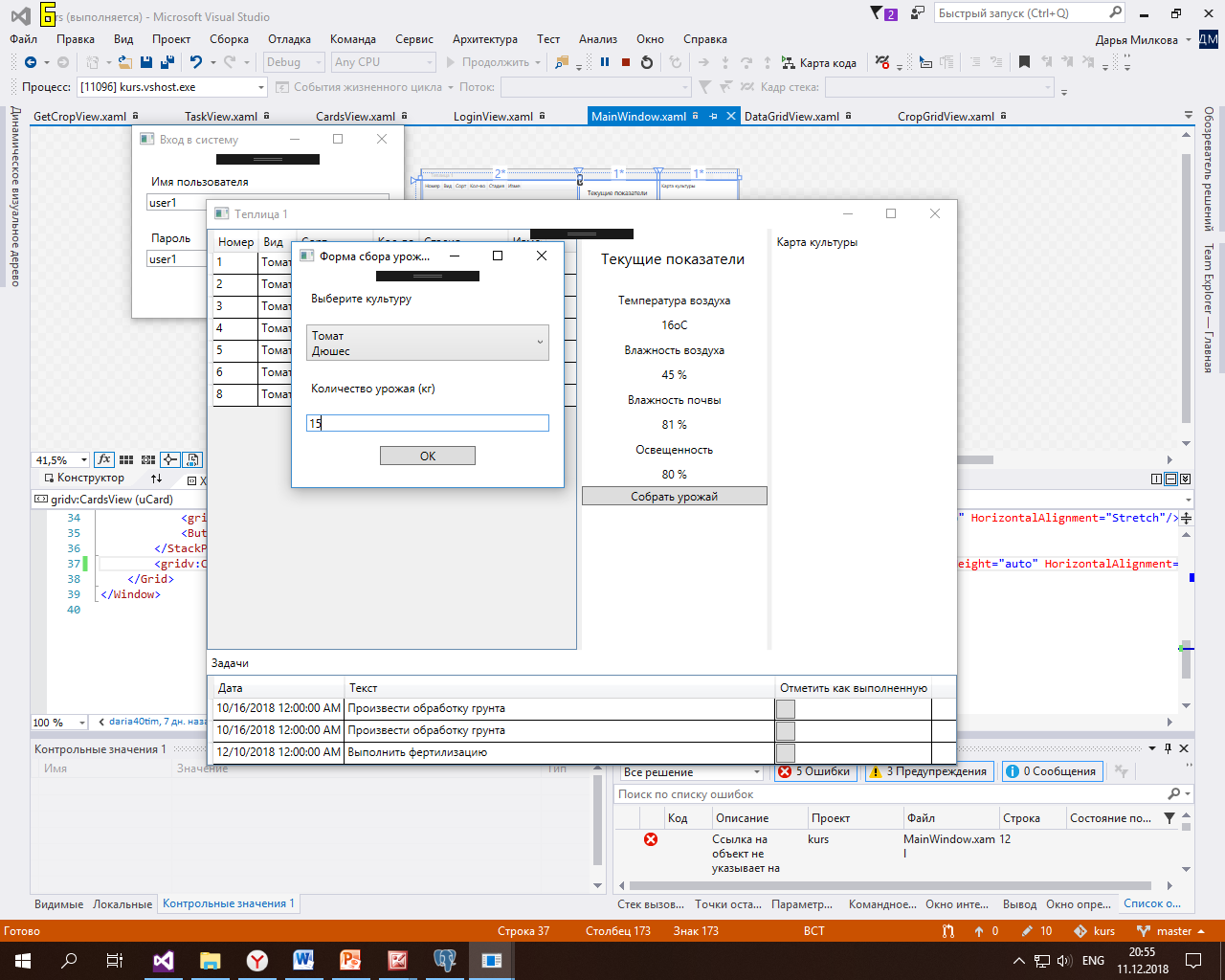


Рисунок 9 – Форма отчета по собранному урожаю

# Разработка архитектуры приложения [5]

Разработка приложения производится с помощью среды программирования MS Visual Studio 2015 на языке программирования C# и средства разработки интерфейса пользователя Windows Presentation Foundation. Хранение данных и управление ими производится с помощью системы управления базами данных PostgreSQL. Связь между уровнями бизнес-логики приложения реализована с помощью NuGet-пакета Npgsql, и предоставляемыми инструментами, такими как DataReader и DataAdapter. Приложение реализует паттерн проектирования MVVM и многослойную архитектуру приложения.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

XAML представляет собой язык декларативного описания интерфейса, основанный на XML. Также реализована модель разделения кода и дизайна, позволяющая разделить реализацию графического интерфейса и уровня бизнес-логики приложения. Кроме того, есть встроенная поддержка стилей элементов, а сами элементы легко разделить на элементы управления второго уровня, которые, в свою очередь, разделяются до уровня векторных фигур и свойств/действий.

Model-View-ViewModel (MVVM) — шаблон проектирования архитектуры приложения. Представлен в 2005 году Джоном Госсманом (John Gossman) как модификация шаблона Presentation Model. Шаблон MVVM делится на три части:

* *Модель* представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения.
* *Представление* — графический интерфейс (окна, списки, кнопки и т. п.). Выступает подписчиком на событие изменения значений свойств или команд, предоставляемых Моделью Представления. В случае, если в Модели Представления изменилось какое-либо свойство, то она оповещает всех подписчиков об этом, и Представление, в свою очередь, запрашивает обновлённое значение свойства из Модели Представления. В случае, если пользователь воздействует на какой-либо элемент интерфейса, Представление вызывает соответствующую команду, предоставленную Моделью Представления.
* *Модель Представления* — с одной стороны, абстракция Представления, а с другой — обёртка данных из Модели, подлежащие связыванию. То есть, она содержит Модель, преобразованную к Представлению, а также команды, которыми может пользоваться Представление, чтобы влиять на Модель.

PostgreSQL - это объектно-реляционная система управления базами данных (СУБД), которая имеет традиционные возможности коммерческих СУБД с расширениями, которые есть в СУБД нового поколения. PostgreSQL - это свободное и полностью открытое программное обеспечение.

# Проектно-техническая часть

# Проектирование начального и тестового наполнения базы данных [3, 8]

Данные о работниках поступают в систему из отдела кадров, оптимизация работы которого в задачи приложения не входила. Поэтому в начальном наполнении базы данных присутствует список работников, уже работающих в теплицах.

Тестовое наполнение базы данных ее физическая схема представлены в приложении.

# Технологические решения, поддерживающие эксплуатационный цикл программы [8]

Проект был разработан с применением подхода database-first, поэтому имеющаяся в нем структура классов соответствует разработанной структуре базы данных и с помощью инструментов, предоставляемых Nu-Get пакетом NpgSQL, легко и эффективно взаимодействует с ней.

Данные из базы данных извлекаются по мере необходимости, после чего соединение закрывается, что также обеспечивает наибольшую эффективность работы, а также быстродействие системы в целом. Разработанная система является стабильной во время своего жизненного цикла, так как в системе реализованы алгоритмы добавления новых культур и задач, что значительно повышает адаптируемость системы к изменяющимся обстоятельствам.

Одним из ключевых достоинств СУБД PostgreSQL является надежность хранения данных, что очень полезно в системах, ге возможен анализ и обработка полученных данных.

# Заключение

В ходе выполнения курсовой работы было реализовано работоспособное программное приложение, соответствующее теме курсовой работы и техническому заданию, оформлен комплект технической документации, а также сформирована графическая часть, а также выполнены следующие задачи:

* + Приобретены навыки системного анализа и моделирования, ведения научно-исследовательской, проектно-конструкторской и проектно-технологической деятельности.
  + Разработано работоспособное программное приложение, автоматизирующее деятельность тепличного комплекса.
  + Разработан комплект технической документации к проекту

При выполнении курсовой работы получены навыки:

* поиска и обработки необходимой информации из различных источников с использованием современных информационных технологий;
* генерации различных вариантов решений поставленных задач и выбора наиболее эффективных вариантов;
* использования методов инфологического проектирования и разработки схем баз данных;
* написания и отладки запросов на языке SQL;
* навыками интеграции алгоритмов работы с СУБД в приложения
* оформления отчетов и технической документации по вопросам проектирования БД и приложений
* представления результатов своей работы в виде доклада;
* защиты полученных результатов в дискуссии.

# Список использованных источников

1. Кудрявцев, К.Я. Создание баз данных: учебное пособие — М.: НИЯУ МИФИ, 2010. — 155 с. — Режим доступа: <http://e.lanbook.com/book/75822>
2. [Мэтью Мак-Дональд. WPF: Windows Presentation Foundation в .NET 4.0 с примерами на C# 2010](https://www.twirpx.com/file/814794/) СПб.: Вильямс, 2011. - 1020с.
3. Ревунков, Г.И. Базы и банки данных - М.: МГТУ им. Н.Э. Баумана, 2011. — 68 с. — Режим доступа: <http://e.lanbook.com/book/52425>
4. Ревунков, Г.И. Проектирование баз данных - М.: МГТУ им. Н.Э. Баумана, 2009. — 20 с. — Режим доступа: <http://e.lanbook.com/book/52390>
5. Рихтер Дж. CLR via C#: Программирование на платформе Microsoft .NET Framework 4.5 на языке C# СПб.: Питер, 2013. — 896 с. — (Серия «Мастер-класс»). — 4-е издание
6. Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификационной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.
7. Шнырёв, С.Л.  Базы данных: учебное пособие для вузов - М. : НИЯУ МИФИ, 2011. — 224 с. — Режим доступа: <http://e.lanbook.com/book/75809>
8. Документация к PostgreSQL 9.6.11

# ПРИЛОЖЕНИЕ

|  |
| --- |
| create table errors( |
|  | e\_id serial primary key, |
|  | text varchar(50), |
|  | data\_of\_err timestamp); |
|  |  |
|  | CREATE TABLE sorts( |
|  | sort\_id serial primary key, |
|  | s\_title varchar(30)); |
|  |  |
|  |  |
|  | CREATE TABLE stages( |
|  | stage\_id serial primary key, |
|  | st\_title varchar(30)); |
|  |  |
|  | CREATE TABLE types( |
|  | type\_id serial primary key, |
|  | t\_title varchar(30)); |
|  |  |
|  | CREATE TABLE greenhouses |
|  | ( |
|  | house\_id serial primary key , |
|  | number integer, |
|  | area integer NOT NULL); |
|  |  |
|  | create table cultures( |
|  | cult\_id serial primary key, |
|  | sort\_id integer references sorts(sort\_id) on delete set null on update cascade, |
|  | type\_id integer references types(type\_id) on update cascade on delete set null, |
|  | house\_id integer references greenhouses(house\_id) on update cascade on delete set null); |
|  |  |
|  | create table workers( |
|  | worker\_id serial primary key, |
|  | fio varchar(50), |
|  | login varchar(20) unique, |
|  | password varchar(20) unique, |
|  | position varchar(20), |
|  | house\_id integer references greenhouses(house\_id) on delete set default on update cascade); |
|  |  |
|  | create table plants( |
|  | p\_id serial primary key, |
|  | house\_id integer references greenhouses(house\_id) on delete set default on update cascade, |
|  | stage\_id integer references stages(stage\_id) on delete set default on update cascade, |
|  | cult\_id integer, |
|  | count integer); |
|  |  |
|  | create table cards( |
|  | card\_id serial primary key, |
|  | card\_title varchar(50), |
|  | stage\_id integer references stages(stage\_id) on delete set default on update cascade, |
|  | cult\_id integer references cultures(cult\_id) on delete set default on update cascade, |
|  | optimal real, |
|  | tolerance real, |
|  | limit\_dev real |
|  | ); |
|  |  |
|  | create table tasks( |
|  | task\_id serial primary key, |
|  | task\_text text, |
|  | worker\_id integer references workers(worker\_id) on delete set default on update cascade, |
|  | day\_of\_app date, |
|  | day\_of\_ending date, |
|  | result varchar(50) |
|  | ); |
|  |  |
|  |  |
|  | create table crops( |
|  | crop\_id serial primary key, |
|  | cult\_id integer references cultures(cult\_id) on update cascade, |
|  | value integer, |
|  | data\_of\_crop date |
|  | ); |
|  | begin; |
|  | insert into stages values |
|  | (default, 'Рассада'), |
|  | (default, 'Рост'), |
|  | (default, 'Цветение'), |
|  | (default, 'Плодоношение') |
|  | returning \*; |
|  | end; |
|  |  |
|  | begin; |
|  | insert into sorts values |
|  | (default, 'Томат'), |
|  | (default, 'Огурец') |
|  | returning \*; |
|  | end; |
|  |  |
|  | begin; |
|  | insert into types values |
|  | (default, 'Дюшес'), |
|  | (default, 'Розовый мед'), |
|  | (default, 'Sweet cherry'), |
|  | (default, 'Герман F1'), |
|  | (default, 'Артист F1') |
|  | returning \*; |
|  | end; |
|  |  |
|  | begin; |
|  | insert into greenhouses values |
|  | (default, 1, 2500), |
|  | (default,2,1000), |
|  | (default, 3,1000) |
|  | returning \*; |
|  | end; |
|  |  |
|  | begin; |
|  | insert into cultures values |
|  | (default, 1,1, 1), |
|  | (default,1,2,1), |
|  | (default, 1,3,1), |
|  | (default, 2,4,2), |
|  | (default, 2,5,3) |
|  | returning \*; |
|  | end; |
|  |  |
|  | begin; |
|  | insert into plants values |
|  | (default,1,2,1,25), |
|  | (default,1,3,1,50), |
|  | (default,1,4,1,50), |
|  | (default,1,3,2,50), |
|  | (default,1,4,3,50), |
|  | (default,1,4,2,50), |
|  | (default,1,4,3,25) |
|  | returning \*; |
|  | end; |
|  |  |
|  | begin; |
|  | insert into tasks values |
|  | (default,'Произвести обработку грунта',null, now(), null, null), |
|  | (default,'Произвести обработку грунта',null, now(), null, null) |
|  | returning \*; |
|  | end; |
|  |  |
|  | begin; |
|  | insert into cards values |
|  | (default,'Температура воздуха', 4,2,15,2,15), |
|  | (default,'Влажность почвы', 2,1,90,5,15), |
|  | (default,'Влажность воздуха', 4,3,45,5,15), |
|  | (default,'Освещенность', 3,2,65,5,15) |
|  | returning \*; |
|  | end; |