**Главное управление по образованию Витебского облисполкома**

**Учреждение образование   
«Полоцкий государственный экономический колледж»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА   
К КУРСОВОМУ ПРОЕКТУ**

**по дисциплине «Базы данных  
 и системы управления базами данных»**

Тема: «Игровое приложение «Эко-ферма». Модуль управления данными»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  |  |
| учащийся 4 курса группы П41  специальности 2-40 01 01 «Программное обеспечение  информационных технологий» | \_\_\_\_\_\_\_\_\_\_\_  (подпись) | Н.В. Сятковский |
| Руководитель: |  |  |
| преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | И.К.Федорчук |

Дата представления\_\_\_.\_\_.\_\_\_

Отметка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Полоцк 2022

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc120746484)

[1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ 6](#_Toc120746485)

[1.1 Анализ предметной области 6](#_Toc120746486)

[1.2 Анализ аналогов и прототипов 8](#_Toc120746487)

[1.3 Разработка технического задания 10](#_Toc120746490)

[1.4 Выводы по первой главе 14](#_Toc120746491)

[2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 15](#_Toc120746492)

[2.1 Проектирование логической модели данных 15](#_Toc120746493)

[2.2 Проектирование физической модели данных 16](#_Toc120746494)

[2.3 Проектирование интерфейса разрабатываемого приложения 18](#_Toc120746495)

[2.4 Выводы по второй главе 22](#_Toc120746498)

[3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА 23](#_Toc120746499)

[3.1 Описание реализации программного средства 23](#_Toc120746500)

[3.2 Тестирование программного средства 23](#_Toc120746501)

[3.3 Установка и эксплуатация ПО 24](#_Toc120746502)

[3.4 Выводы по третьей главе 24](#_Toc120746503)

[ЗАКЛЮЧЕНИЕ 25](#_Toc120746504)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](#_Toc120746505)

[ПРИЛОЖЕНИЕ А Диаграмма вариантов использования 27](#_Toc120746506)

[ПРИЛОЖЕНИЕ Б Диаграмма сущность-связь 28](#_Toc120746507)

[ПРИЛОЖЕНИЕ В Техническое задание 28](#_Toc120746508)

[ПРИЛОЖЕНИЕ Г Руководство оператора 32](#_Toc120746509)

[ПРИЛОЖЕНИЕ Д Программа и методика испытаний 38](#_Toc120746510)

[ПРИЛОЖЕНИЕ EСодержание электронного носителя 43](#_Toc120746511)

# ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

|  |  |
| --- | --- |
| ОР | – Объект развития |

# ВВЕДЕНИЕ

Информационная система – это система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы, которые обеспечивают и распространяют информацию. Информационные системы становятся подходящим способом долговременного хранения и обработки информации.

Сегодня управление предприятием без компьютера просто немыслимо. Любая организация нуждается в своевременном доступе к информации. Ценность информации в современном мире очень высока. Роль распорядителей информации в современном мире чаще всего выполняют базы данных. Базы данных обеспечивают надежное хранение информации в структурированном виде и своевременным доступ к ней.

Актуальность данной темы в том, что в наш век информационных технологий, стало реально все документы преобразовывать в электронный вид и администратор в считанные минуты может найти информацию о банкете.

Объектом курсового проекта является автоматизация формирования каталога игровых сущностей и статистики игрового процесса.

Предметом курсового проектирования является игровое приложение «Эко-ферма». Модуль управления данными.

Целью курсового проектирования является разработка программного средства предназначенное для автоматизации формирования каталога игровых сущностей и статистики игрового процесса.

Для реализации цели курсового проектирования необходимо решить следующие задачи:

* изучить предметную область;
* проанализировать функциональные требования к разрабатываемому приложению;
* определить инструменты для реализации проекта;
* создать базу данных для приложения;
* спроектировать интерфейс приложения;
* спроектировать приложение;
* реализовать приложение;
* осуществить тестирование созданного приложения;
* разработать необходимую эксплуатационную документацию.

Разработанное приложение представляет собой файл-серверную систему. Эксплуатируемой базой данных является SQLite v3.40.0. Языком реализации приложения является C#, средой программирования является JetBrains Rider.

# АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ

## Анализ предметной области

Предметной областью, для которой ведется разработка является индустрия компьютерных игр. [5]

Индустрия компьютерных игр – сектор экономики, связанный с разработкой, продвижением и продажей компьютерных игр. В неё входит большое количество специальностей, по которым работают тысячи человек по всему миру.

Бен Сойер из Digitalmill рассматривает цепочку ценности игровой индустрии, которая составлена из шести связанных, но хорошо различимых слоев:

– уровень издательства и капитала: издательства оплачивают разработку новых проектов и извлекают прибыль посредством лицензирования наименований;

– уровень продукта и талантов: включает в себя разработчиков, дизайнеров, художников, композиторов, которые могут работать как по индивидуальным контрактам, так и в составе группы разработчиков;

– уровень создания и технологий: является источником средств разработки игр, настраиваемых и расширяемых игровых движков, связующего ПО, а также средств управления разработкой;

– уровень распространения: создание перечней игр и их продвижение в розничных и интернет-магазинах;

– уровень аппаратного и программного обеспечения: сюда входят аппаратные базисы-платформы, в том числе такие как консоли и мобильные устройства. В этот уровень сейчас входят также и неаппаратные платформы, такие как виртуальные машины или программные платформы, такие как браузеры или Facebook;

– уровень конечных пользователей. Иначе говоря, потребителей игр или геймеров.

Одновременно с развитием технологий происходило и развитие рынка игр. Так в начале периода стали более успешными лицензированные игры, также как и сиквелы игр.

В 1993 году продажи компьютерных игр в мире составили $19,8 млрд. ($31 млрд. в ценах 2011), $20,8 млрд. в 1994 ($32 млрд. в ценах 2011) и приблизительно $30 млрд. в 1998 ($41,5 млрд. в ценах 2011). Суммарные продажи игровой индустрии США более чем в 2,5 раза превысили продажи кинематографа в США.

В 2000-е было создано и стали популярными множество казуальных и инди-игр. Также крепнет направление игр для мобильных платформ, появилось направление создания игр для социальных сетей. Особенно известен разработчик Zynga игр для социальной сети Facebook. Другим примером успешных платформ для компьютерных игр являются iOS и Android.

На начальном этапе существования компьютерных игр стоимость разработки была минимальной, поэтому это был прибыльный бизнес. Игры, разработанные единственным программистом или небольшой группой, состоящей из программиста и нескольких художников, могли обеспечивать продажи в количестве сотен тысяч копий. Многие из этих игр были разработаны всего за несколько месяцев, что давало возможность разработчикам выпускать по несколько игр в год. Это давало возможность издателям предлагать весьма щедрые отчисления разработчикам, включая различные роялти с проданных копий. В течение этого экономически благоприятного периода было создано много известных компаний-издателей, например, Origin Systems, Sierra Entertainment, Capcom, Activision и Electronic Arts.

На данный момент игровая библиотека только в одном онлайн-сервисе цифрового распространения игр Steam опубликованы десятки тысяч игр от тысяч разработчиков со всего мира.

В настоящее время компьютерные игр вносят значительный вклад в мировую экономику ввиду большого успеха продаж основных игровых систем и игр типа Call of Duty: Black Ops, заработавшая в течение первых 5 дней продаж более $600 млн., что стало мировым рекордом пятидневных продаж среди фильмов, книг и компьютерных игр.

В последние годы набирает популярность модель «пожертвований» (с помощью сервисов аналогичных Kickstarter) для разработки компьютерных игр. Часто известные в прошлом разработчики игр предлагают создать идейное «продолжение» или «аналог» известных игр, если пользователи соберут запрашиваемую сумму. Примерами проектов, наиболее успешно собравших средства с помощью этой схемы, являются Star Citizen (запросили – $2 млн., собрали – $27 млн.), Torment: Tides of Numenera (запросили – $1 млн., собрали – более $4,2 млн.) и Elite: Dangerous (запросили – $1,25 млн., собрали – более $1,7 млн.).

Каждая игра имеет уникальный игровой процесс(геймплей) и относится к одному из игровых жанров: platformer, CCG, shooter, RPG, simulator, MOBA, quest, arcade, racing, puzzle, sandbox, horror и так далее. Они помогают игрокам примерно понимать, чего стоит ожидать от данной конкретной игры. Ведь даже, казалось бы, схожие «аркадные гонки» и «гоночные симуляторы» на самом деле обеспечивают разный игровой опыт, в то время как первые своей целью ставят непосредственно весёлое времяпрепровождение с друзьями и нередко внедряют в игровой процесс всяческие бонусы дабы разнообразить игровой процесс, вторые делают упор на максимально правдоподобную симуляцию опыта от настоящего вождения.

За свой, небольшой, относительно других сфер искусства, срок жизни в игровой индустрии сменилось 8 поколений игровых консолей, на данный момент актуальными являются 5 основных игровых платформ: PC, PlayStation, Nintendo Switch, Xbox и mobile, однако до сих пор можно приобрести и сыграть на устаревших платформах вроде NES или 3DS. Игры могут выпускаться как эксклюзивно для определённой платформы, так и на нескольких сразу.

Немалую долю рынка занимают проекты с упором на онлайн составляющую (мультиплеер). Однако не стоит забывать, что разработка как крупнейших проектов ААА класса, так и мелких indie игр зачастую стоит немалых денег и каждому разработчику приходится самостоятельно определять каким образом заработать денег с игры. Исторически самым первым методом распространения игр является непосредственная их покупка игроками, однако стоит отметить, что игрок, покупая игру, получает её в бессрочную аренду, но не получает полного владения ней, то есть не может создавать цифровые копии купленной игры и продавать их другим людям. Но с развитием интернета стали появляться и игры, которые распространяются по модели «Free to Play», их можно официально скачать и начать играть абсолютно бесплатно, однако в самой игре может присутствовать контент доступный исключительно по внутри игровой покупке. Данная модель распространения пользуется особой популярностью на мобильном рынке. [6]

## Анализ аналогов и прототипов

Для формирования требований к разрабатываемой программе необходимо изучить аналоги и прототипы.

Аналогами приложения послужили следующие прототипы:

### MS Excel (<https://www.microsoft.com/ru-ru/microsoft-365/excel>).

Microsoft Excel – табличный процессор, разработанный Microsoft для Windows, macOS, Android и iOS. Он включает в себя возможности расчета или вычислений, графические инструменты, сводные таблицы и язык программирования макросов – VBA. Является частью пакета программного обеспечения Microsoft Office (рисунок 1.1). [7]

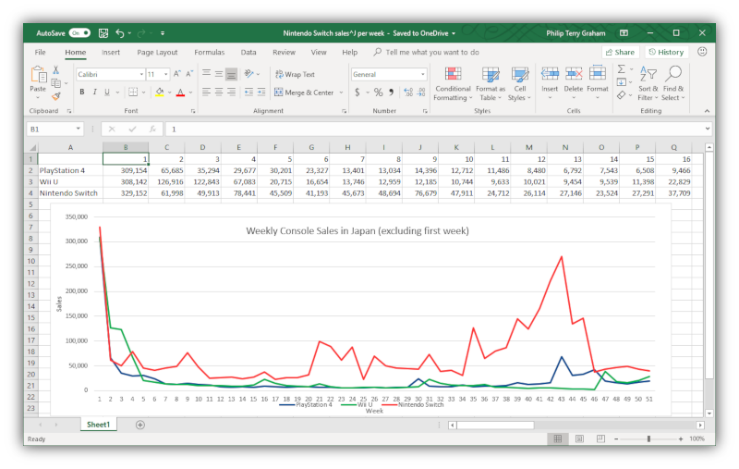


Рисунок 2.1 – Окно программы Microsoft Excel

Достоинства программы:

* мультиплатформенность;
* наличие веб-версии;
* включён в стандартный пакет Microsoft Office;
* синхронизация с облаком.

Недостатки приложения:

* труден в освоении для новых пользователей;
* веб-версия урезана по функционалу;
* высокая стоимость лицензии.

### Google Sheets (<https://www.google.com/sheets/about/>).

Google Sheets – табличный процессор, входящий в состав бесплатного набора веб-редакторов Google Docs Editors, предлагаемого Google. Google Sheets доступна в виде веб-приложения, мобильного приложения для: Android, iOS и настольного приложения в Google ChromeOS (рисунок 1.2).

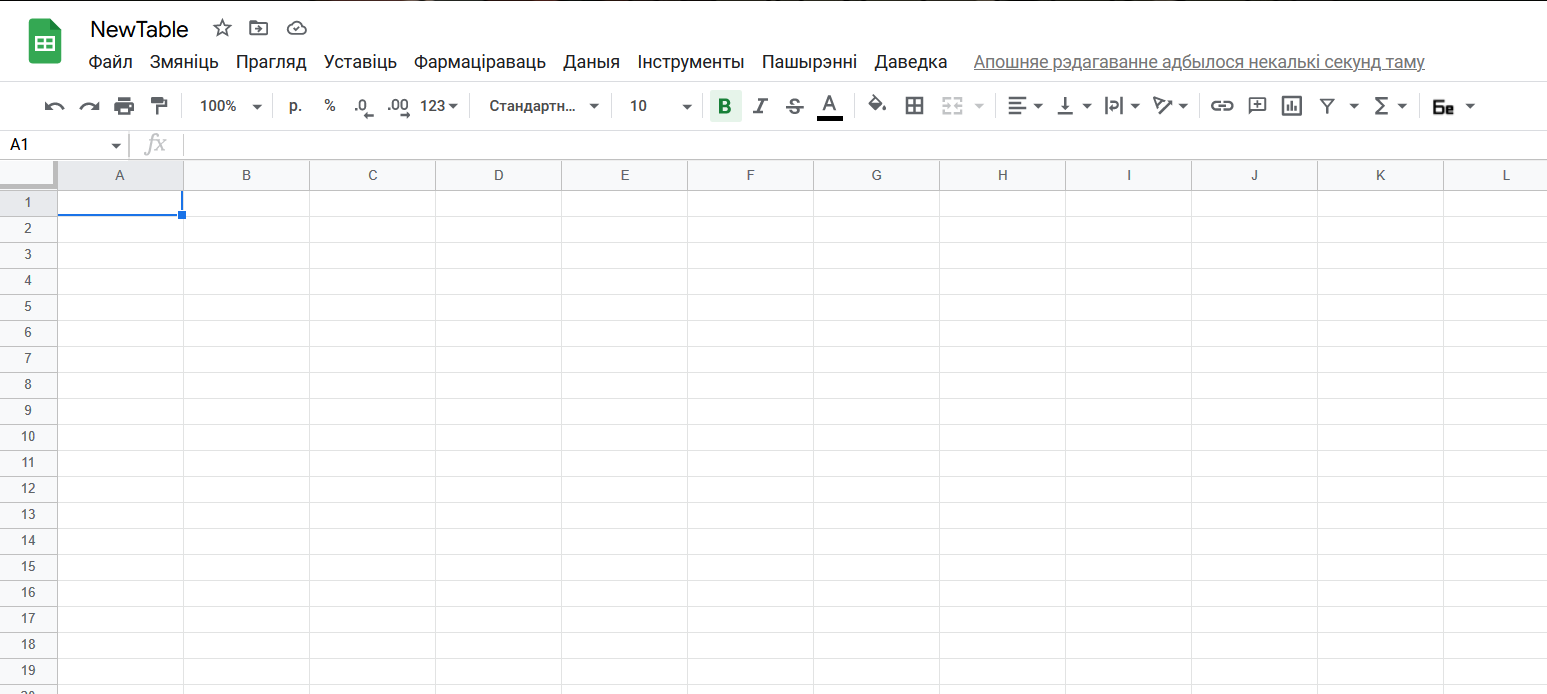


Рисунок 2.2 – Окно программы Google Sheets

Достоинства приложения:

* удобная веб-версия;
* мобильные версии;
* широкие возможности для совместной разработки;
* высокая популярность в управлении игровыми данными;
* бесплатный доступ для пользователя;
* синхронизация с облаком.

Недостатки приложения:

* необходимо думать над структурой данных;
* отсутствие настольной версии для Windows, Linux, Mac.

Проанализировав все аналоги, можно сделать выводы: каждое программное средство имеет свои достоинства и недостатки. Похожих аналогов данной программы не существует, в связи с этим разрабатываться система будет на основе приведённых выше систем. Учитывая их особенности и недостатки, программа «Эко-ферма. Модуль управления данными» будет иметь преимущество перед конкурента в силу своей индивидуальности и предлагаемым требованиями:

* бесплатное использование программы;
* высокая скорость работы;
* простой и понятный интерфейс.

## Разработка технического задания

В данном пункте представлено описание назначения программного средства, его функциональность, эксплуатационные и технические требования.

**1.3.1 Спецификация функций программного средства**

Диаграммы вариантов использования описывают взаимодействия и зависимости между группами вариантов использования и действующих лиц, участвующими в процессе. [8]

Для данного приложения, в соответствии с диаграммой вариантов использования (ПРИЛОЖЕНИЕ А), необходима реализация следующих функций для роли администратора:

Администратор:

* **формирование списков объектов;**
* формирование списков зданий;
* формирование списков игрового процесса
* **экспорт в Json**.

**1.3.2 Обоснование архитектуры программного средства и способа хранения данных**

Для создания информационной системы можно использовать различные архитектуры. Рассмотрим архитектуру СУБД и её виды.

Системы управления базами данных (СУБД) – это программные средства, предназначенные для создания, наполнения, обновления и удаления баз данных.

По своей архитектуре СУБД делятся на одно-, двух- и трехзвенные. В однозвенной архитектуре используется единственное звено (клиент), обеспечиваю­щее необходимую логику управления данными и их визуализацию. В двухзвенной архитектуре значительную часть логики управления данными берет на себя сервер БД, в то время как клиент в основном занят отображением данных в удобном для пользователя виде. В трехзвенных СУБД используется промежуточное звено – сервер приложений, являющееся посредником между клиентом и серве­ром БД. Сервер приложений призван полностью избавить клиента, от каких бы то ни было забот по управлению данными и обеспечению связи с сервером БД.

В архитектуре «клиент-сервер» сервер базы данных не только обеспечивает доступ к общим данным, но и берет на себя всю обработку этих данных. Клиент посылает на сервер запросы на чтение или изменение данных, которые формулируются на языке SQL. Сервер сам выполняет все необходимые изменения или выборки, контролируя при этом целостность и согласованность данных, и результаты в виде набора записей или кода возврата посылает на компьютер клиента.

**Архитектура данной программы локальная СУБД, так как все данные хранятся в определённом разделе в реестре компьютера пользователя.**

**1.3.3 Обоснование выбора СУБД**

Для хранения данных и их синхронизации для всех пользователей в системе должны использоваться системы управления базами данных.

Были расмотрены следующие варианты СУБД: Microsoft SQL Server, SQLite.

Microsoft SQL Server – СУБД, разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase.

Достоинство SQL Server:

* рутинные административные задачи автоматизированы: это управление блокировками, памятью, редактура размеров файлов;
* реализован поиск по фразам, тексту, словам, можно создавать ключевые индексы;
* в систему интегрирован сервер интерактивного анализа для принятия решений, создания корпоративных отчетов;
* запросы можно формулировать на английском языке, без программирования;
* СУБД поддерживает работу с другими продуктами Microsoft: Access, MS Excel.

SQLite – это механизм базы данных, написанный на языке программирования C. Это не отдельное приложение; скорее, библиотека, которую разработчики программного обеспечения встраивают в свои приложения. Таким образом, он принадлежит к семейству встроенных баз данных. Это наиболее широко распространенный движок базы данных, так как он используется несколькими ведущими веб-браузерами, операционными системами, мобильными телефонами и другими встроенными системами.

Достоинства SQLite:

* Высокая скорость работы;
* Хранение данных в одном файле;
* Надежность;
* Доступность;
* Кроссплатформенность;
* Автономность;
* Малый размер.

Для разработки программы «Эко-ферма. Модуль управления данным» использовалось СУБД SQLite, поскольку данная СУБД легко устанавливается для создания файл-серверной архитектуры и имеет интеграцию с Unity. [9]

**1.3.4 Обоснование выбора языка программирования**

Игра была разработана на Unity, который поддерживает скрипты на языке C#. Для экспорта данных в неё – необходима единая библиотека для сериализации и десериализации игровых объектов, следовательно программа должна разрабатываться на языке программирования C#.

Для разработки программы «Эко-ферма. Модуль управления данным» использовался язык программирования C#.

Язык программирования C# является объектно-ориентированным. Имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Разработан в 1998 – 2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. Позволяет в автоматическом режиме очистить память от объектов, которые не используются, или уничтоженных приложений. В языке принята общая система работы с типами, начиная от примитивов и заканчивая сложными, в том числе, пользовательскими наборами. Применяется единый набор операций для обработки и хранения значений типизации. Также можно использовать ссылочные типы пользователя, что позволит динамически выделить память под объект или хранить упрощенную структуру в сети.

В процессе выбора технологии работы с базой данных были рассмотрены:

* Entity Framework;
* Dapper;
* ADO.NET.

Entity Framework – объектно-ориентированная технология доступа к данным, является object-relational mapping решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL.

Dapper – технология сопоставления (маппинга) результатов sql-запросов с классами c#. В этом плане Dapper немного похож на Entity Framework. В то же время за счет своей легковесности Dapper обеспечивает большую производительность и быстрее позволяет выполнять запросы, нежели Entity Framework.

ADO.NET – это технология доступа к данным из Microsoft .NET Framework, которая обеспечивает связь между реляционными и нереляционными системами через общий набор компонентов.

Основной средой разработки была выбрана JetBrains Rider ввиду ее функциональности, надежности и предпочтения разработчика.

Rider – кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#, VB.NET и F#.

Для разрабатываемого приложения была выбрана технология Entity Framework. Так как о позваляет работать с БД, как с набором обычным C# классов, что значительно повышает скорость разработки. Для создания приложения выбрана среда разработки JetBrains Rider, так как данная IDE имеет лучшую интеграцию с Unity и инструменты для работы с WPF, чем Visual Studio.

**1.3.5 Обоснование эксплуатационных требований**

Системные требования составлялись исходя из конфигурации ПК, на котором происходили разработка и тестирования ПО.

Разрабатываемое программное средство предполагает эксплуатацию под операционной системной не меньше Windows 10, программная платформа NET. Framework 4.8.

Минимальные аппаратные требования к компьютеру пользователя следующие:

– процессор: «Intel Core 2 Duo» и новее;

– ОЗУ: 4 ГБ;

– место на жестком диске: не менее 500 МБ;

– видеоадаптер: DirectX 9 или более поздняя версия.

## Выводы по первому разделу

Исходя из вышеперечисленной информации сформирован вывод для данной главы:

* была проанализирована предметная область игровой индустрии;
* были проанализированы аналоги и прототипыы;
* были разработаны функциональные требования к системе;
* были проанализированы нефункциональные требования к системе.
* было разработано техническое задание (ПРИЛОЖЕНИЕ Б).

# ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## Проектирование логической модели данных

Для приложения была разработана диаграмма «Сущность-связь», которая представлена в (ПРИЛОЖЕНИИ В).

На диаграмме выделены следующие сущности: ОР, ОР на старте уровня, уровень, Цель, Цель по ОР, Здание, Выходные продукты, Продукт, Входные продукты, Дерево, Фабрика, Генератор, Ресурс.

Связи в данной диаграмме являются:

* + ОР – уровень: один уровень может иметь несколько ОР и один ОР может быть на нескольких уровнях;
  + Игрок – Уровень: Игрок может пройти несколько уровней и один уровень может быть пройден несколькими игроками;
  + Цель – Уровень: один уровень может содержать одну и более целей;
  + Цель – ОР: одна цель может быть по нескольким ОР, один ОР может содержаться в нескольких целях;
  + Дерево – Продукт: каждое дерево производит один продукт, один продукт может быть произведён разными деревьями;
  + Здание – Производство: каждое производство является зданием, но не каждое здание является производством;
  + Здание – Генератор: каждый генератор является зданием, но не каждое здание является генератором;
  + Производство – Продукты: Одно производство принимает и выдаёт несколько продуктов;
  + Производство – Ресурс: каждое производство может использовать несколько ресурсов, один ресурс может использоваться на нескольких производствах;
  + Генератор – Ресурс: каждый генератор генерирует один ресурс, каждый ресурс может быть сгенерирован несколькими генераторами.

Описание сущностей представлены в таблицах 2.1 – 2.4.

Таблица 2.1 – Структура сущности «Уровень»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_Уровня | Обязательное (ключевое поле) | Первичный ключ, идентификатор |
| Название уровня | Обязательное | Название уровня |
| Описание уровня | Обязательное | Описание уровня |
| Состояние уровня | Обязательное | Состояние уровня |

Таблица 2.2 – Структура сущности «Цель»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_Цели | Обязательное (ключевое поле) | Первичный ключ, идентификатор |
| ID\_Уровня | Обязательное | Внешний ключ, идентификатор |
| Название уровня | Обязательное | Название уровня |
| Описание уровня | Обязательное | Описание уровня |
| Состояние уровня | Обязательное | Состояние уровня |

Таблица 2.3 – Структура сущности «Ресурс»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_Ресурса | Обязательное (ключевое поле) | Первичный ключ, идентификатор |
| Наименование | Обязательное | Наименование |

Таблица 2.4 – Структура сущности «Статистика игры»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_Уровня | Обязательное | Внешний ключ, идентификатор |
| ID\_Цели | Обязательное | Внешний ключ, идентификатор |
| Значение | Обязательное | Значение |

## Проектирование физической модели данных

Физическая модель базы данных содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей и т.п. Создание физической модели данных является последним этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных.

На основании логической модели и выбора инструментальных средств была спроектирована физическая модель базы данных.

Таблицы физической модели данных приложения:

* ОР – хранит информацию об ОР;
* ОР на старте уровня – хранит информацию об ОР на старте уровня;
* Уровень – хранит информацию о уровне;
* Цель – хранит информацию;
* Цель по ОР – хранит информацию об Целях по ОР;
* Здание – хранит информацию о зданиях;
* Выходные продукты – хранит информацию о выходных продутках;
* Продукт – хранит информацию о продухтах;
* Входные продукты – хранит информацию о входных продуктах;
* Дерево – хранит информацию о деревьях;
* Фабрика – хранит информацию о фабриках;
* Генератор – хранит информацию о генераторах;
* Ресурс – хранит информацию о ресурсах.

Подробное описание сущностей представлено в таблицах 2.6 – 2.10.

Таблица 2.6 – Структура сущности «Заказы»

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание атрибута |
| ID\_заказы | integer | Primary key, not null, identify(1,1) |
| ID\_клиенты | integer | Foreign key, not null, identify(1,1) |
| Дата начала | datetime | Not null |
| Время начала | datetime | Not null |
| Количество людей | integer | Not null |

Таблица 2.7 – Структура сущности «Блюда»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_блюда | integer | Primary key, not null, identify(1,1) |
| Название | varchar(20) | Not null |
| Стоимость | money | Not null |
| Объём порции | integer | Not null |
| Время приготовления | datetime | Not null |

Таблица 2.8 – Структура сущности «Блюда в заказе»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_заказы | integer | Primary key, not null, identify(1,1) |
| ID\_блюда | integer | Foreign key, not null, identify(1,1) |
| Количество | integer | Not null |

Таблица 2.9 – Структура сущности «Клиенты»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_клиента | integer | Primary key, not null, identify(1,1) |
| ФИО | varchar(20) | Not null |
| Телефон | integer | Not null |

Таблица 2.10 – Структура сущности «Категория»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_категории | integer | Primary key, not null, identify(1,1) |
| Название | varchar(20) | Not null |

## Проектирование интерфейса разрабатываемого приложения

В данном пункте пояснительной записки представлен шаблон интерфейса программного средства в виде: функций приложения, графов перехода и экранных форм ввода вывода информации.

**2.3.1 Спецификация функций ПО на экранных формах**

В данном пункте пояснительной записке определяются основные функции приложения. Функции приложения представлены в таблице 2.11, а перечень экранных форм представлена в таблице 2.12.

Таблица 2.11 – Перечень функций приложения

|  |  |  |  |
| --- | --- | --- | --- |
| № функции | Название функции | Роль | Перечень форм |
| 1 | 2 | 3 | 4 |
| 1 | Формирование списка продуктов | Администратор | N2 |
| 2 | Формирование списка деревьев | Администратор | N3 |
| 3 | Формирование списка генераторов | Администратор | N4 |
| 4 | Формирование списка фабрик | Администратор | N5 |
| 5 | Формирование списка уровней | Администратор | N6 |
| 6 | Формирование списка ресурсов | Администратор | N7 |
| 7 | Экспорт в Json | Администратор | N8 |

Таблица 2.12 – Перечень экранных форм

|  |  |
| --- | --- |
| № формы | Название |
| N1 | Главное окно |
| N2 | Список продуктов |
| N3 | Список деревьев |
| N4 | Список генераторов |
| N5 | Список фабрик |
| N6 | Список уровней |
| N7 | Список ресурсов |
| N8 | Экспорт |

### Граф переходов состояний интерфейса

В данном пункте представлены граф переходов по экранным формам для администратора.

Граф переходов для администратора показан на рисунке 2.1.

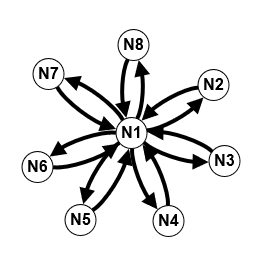


Рисунок 2.1 – Граф переходов «Администратор»

### 2.3.3 Проект экранных форм

Для приложения был разработан шаблон пользовательского интерфейса в программе Microsoft Blend.

Данный интерфейс представлен на рисунках 2.2 – 2.9.

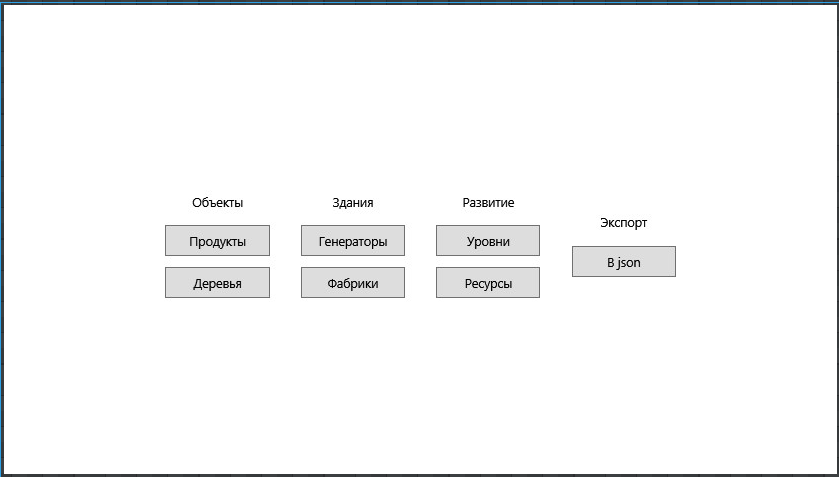


Рисунок 2.2 – Главное окно

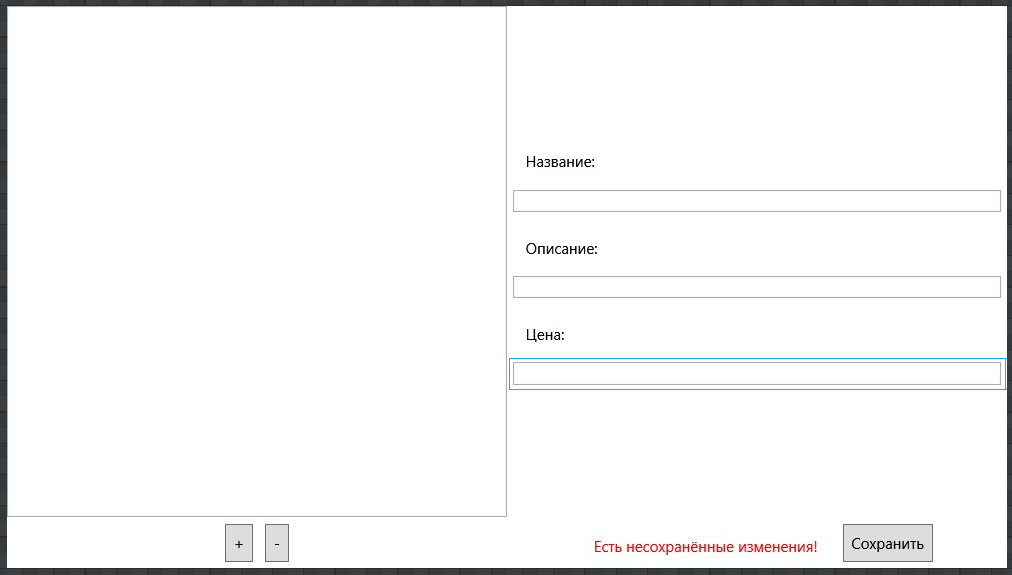


Рисунок 2.3 – Список продуктов

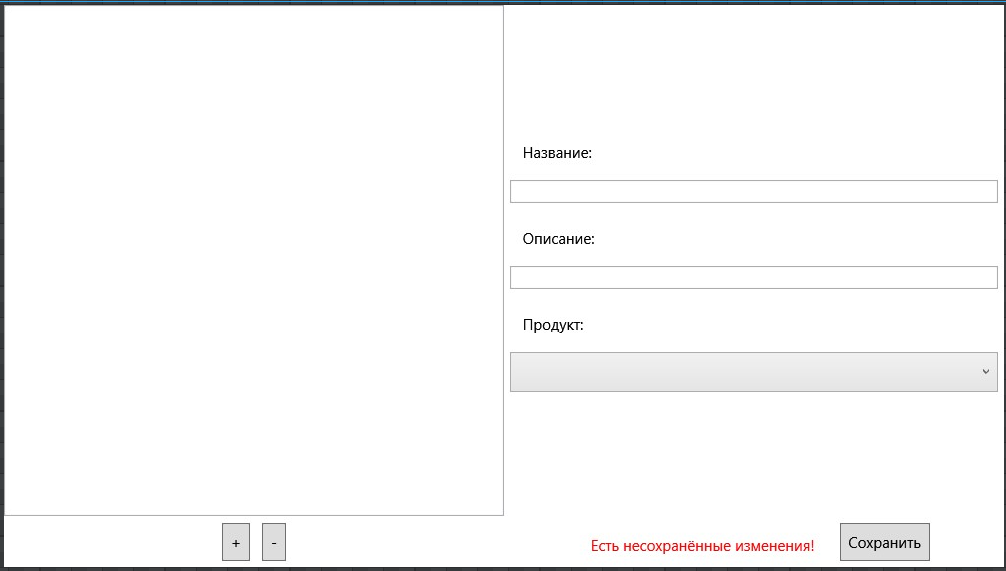


Рисунок 2.4 – Список деревьев

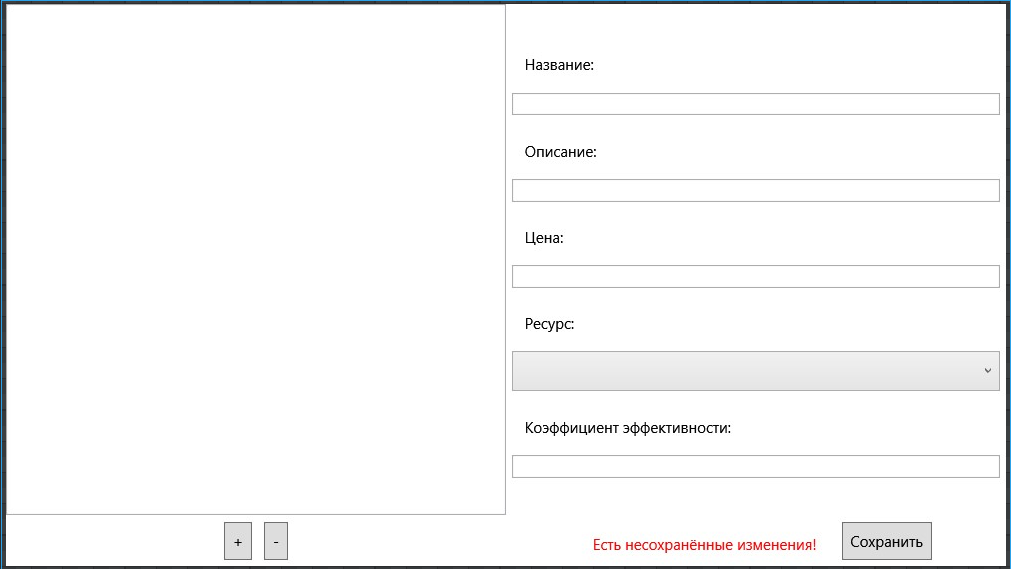


Рисунок 2.5 – Список генераторов

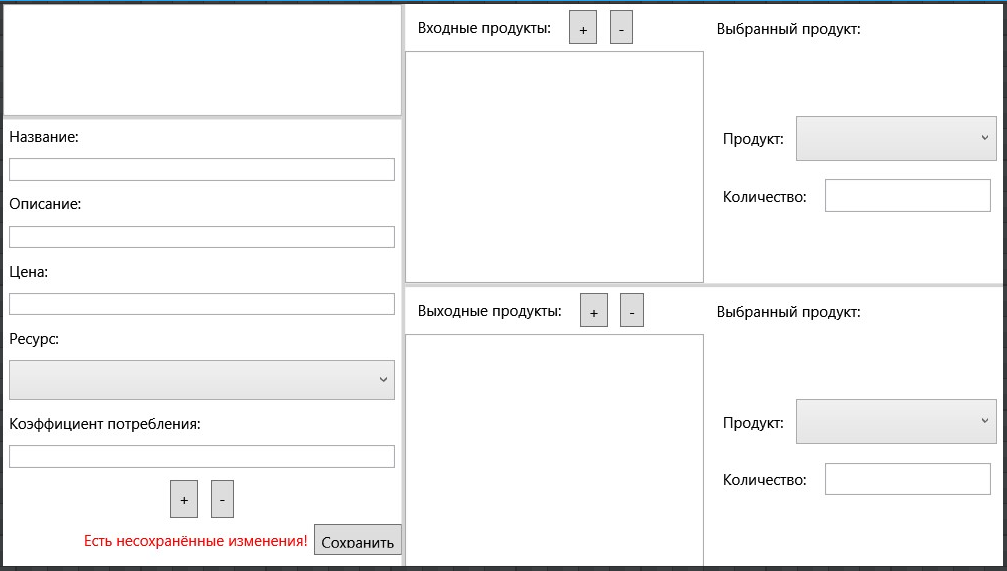


Рисунок 2.6 – Список фабрик

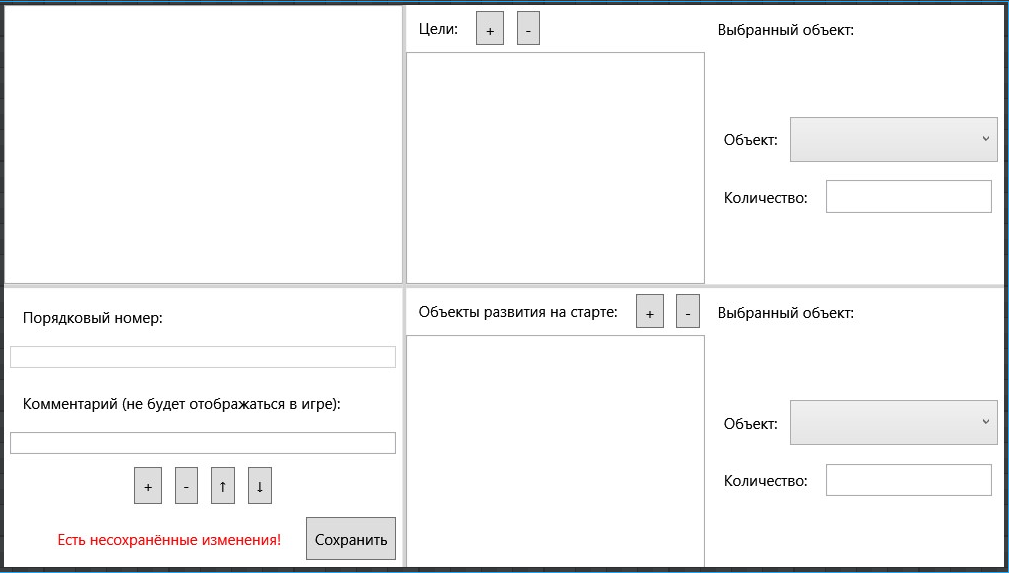


Рисунок 2.7 – Список уровней

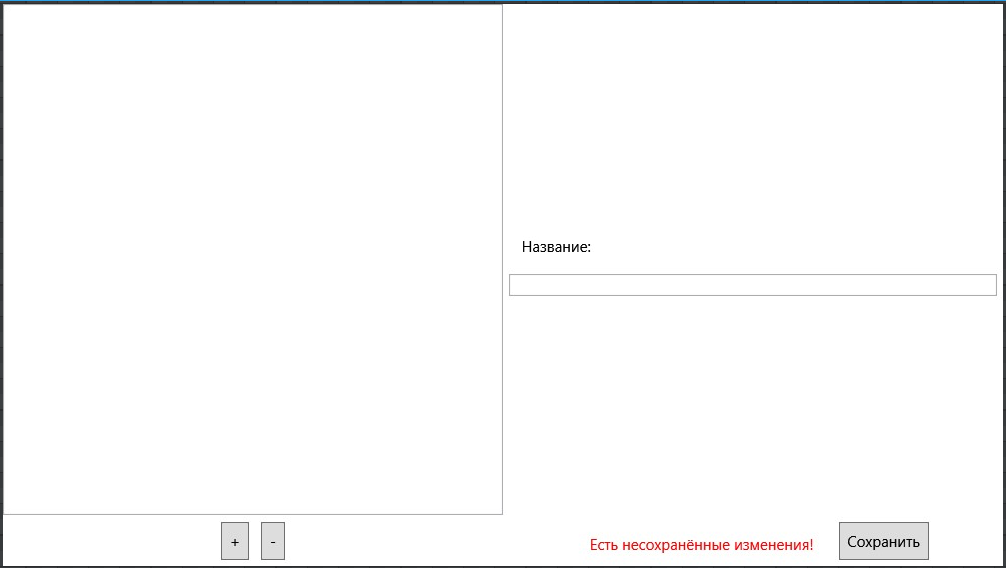


Рисунок 2.8 – Список ресурсов

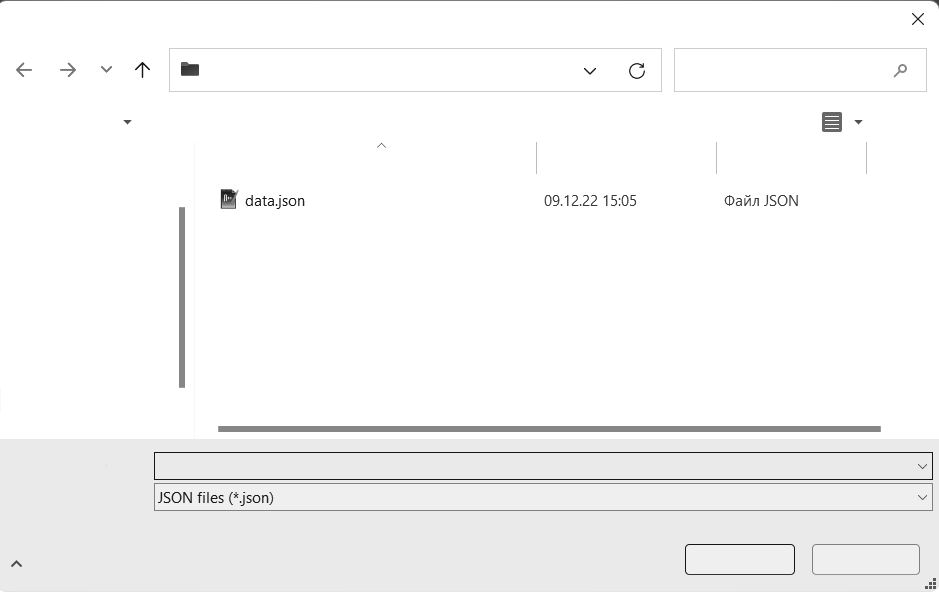


Рисунок 2.9 – Окно экпорта

## Выводы по второму разделу

Исходя из вышеперечисленной информации сформирован вывод для данного раздела:

* была спроектирована логическая модель данных;
* была спроектирована физическая модель данных;
* была разработана диаграмма «сущность-связь»;
* спроектирован шаблон пользовательского интерфейс.

# РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

## Описание реализации программного средства

### Структура разработанного ПО

В ходе курсового проектирования была разработана программа «Эко-Ферма. Модуль управления данными».

Данная программа состоит из нескольких компонентов: файлов базы данных, библиотек, исполняемого файла. Состав и назначение компонентов подсистемы представлена в таблице 3.1.

Таблица 3.1 – Назначение файлов системы

|  |  |
| --- | --- |
| Имя файла | Назначение |
|  | Исполняемый файл |
|  | Файл, содержащий отладочную информацию, необходимую для исполнения программы |
|  | Файл базы данных |

Данные для работы находятся на клиенте. Работа с базой данных осуществляется через технологию маппинга таблиц в C# классы.

Для работы с данной технологией была создан класс-контекст, для работы с БД. Особенность также заключается и в том, что Entity Framework хранит изменения в оперативной памяти и сохранит их только по запросу пользователя.

Логическая структура разработанного программного средства в виде диаграммы развертывания представлена на чертеже 3.08.214124.03 «Диаграмма развертывания» и в ПРИЛОЖЕНИИ Е.

На диаграмме представлена 5 узлов:

1. Сервер базы данных.
2. Рабочая станция администратора.
3. Рабочая станция менеджера.
4. Рабочая станция курьера.
5. Принтер.

На сервере базы данных находятся файлы базы данных: delivery.mdf, delivery \_log.ldf, и СУБД «MS SQL Server 2019».

На рабочих станциях администратора менеджера и курьера находятся одинаковые компоненты, а именно:

1. Исполняемый файл программы «delivery».
2. Библиотека файлов .NET Framework.
3. Файл подключения к базе данных «AppConfig».
4. Библиотека сопоставления результатов SQL-запросов с классами C#.

Принтер используется для вывода квитанций доставок.

### Реализация структуры хранения данных

Структурой хранения данных программы «Эко-Ферма. Модуль управления данными» является база данных, реализованная в СУБД SQLite.

## Тестирование программного средства

Тестирование программного обеспечения – процесс исследования, испытания программного обеспечения (ПО) с целью получения информации о качестве продукта.

С целью исключения возможных ошибок в работе, программа должна пройти функциональное тестирование.

Функциональное тестирование является одним из ключевых видов тестирования, задача которого – установить соответствие разработанного программного обеспечения исходным функциональным требованиям заказчика. Тестирование приложения выполнено согласно документу «Программа и методика испытаний» (ПРИЛОЖЕНИЕ Е).

Результаты функционального тестирования представлены в ПРИЛОЖЕНИИ Д.

В ходе функционального тестирования достаточно большая часть тестов была пройдена успешно. На основании проведенного функционального тестирования НАЗВАНИЕ, можно сделать следующие выводы: программное средство обладает достаточной надежностью.

## Установка и эксплуатация ПО

Эксплуатационное назначение системы – использование администратором базы данных игрового приложения «Эко-ферма».

Для установки программного средства необходимо распаковать файлы программы в любое удобное место на компьютере.

На самом компьютере должна быть установлена СУБД SQLite, к которой подключается файл базы данных, находящийся в папке с программой. После чего программа будет готова к работе.

Процесс эксплуатации программного продукта описан в документе «Руководство оператора» в ПРИЛОЖЕНИИ Л.

## Выводы по третьей главе

В ходе выполнения курсового проекта была разработана НАЗВАНИЕ. Была определена и реализована структура хранения данных. Также было проведено функциональное тестирование. В ходе тестирования программа показала себя на высоком уровне, как надежный продукт.

# ЗАКЛЮЧЕНИЕ

В результате проделанной работы была реализована приложение, способное упростить формирование каталога игровых сущностей.

В ходе разработки данного приложения были изучены основные принципы работы системы, проанализированы и составлены функциональные требования к разрабатываемой системе.

Разработанное приложение было реализовано на языке C#, а его система хранения данных была построена в СУБД SQLite.

Разработанное программное средство удовлетворяют всем выдвинутым в техническом задании требованиям.

При реализации приложения были спроектирована база данных, которая позволяет хранить данные о игровых сущностях.

В процессе выполнения курсового проектировании было выполнено функциональное тестирование приложения в соответствии с документами «Программа и методика испытаний», результаты которого подтвердили, что приложение работает исправно, выполняя все функции, описанные в техническом задании.

Дополнительно были разработаны диаграмма вариантов использования и диаграмма «Сущность-связь», «Руководство оператора».

Результатом разработки стало приложение, которое позволяет формировать каталог игровых сущностей.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению: ГОСТ 19.201–78 ЕСПД. Введ. 01.01.1980 – М.: Издательство стандартов, 1987. – 8 с.
2. Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению: ГОСТ 19.503–79 ЕСПД. Введ. 01.01.1980 – М.: Издательство стандартов, 1982. – 8 с.
3. Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению: ГОСТ 19.505–79 ЕСПД. Введ. 01.01.1980 – М.: Издательство стандартов, 1982. – 8 с.
4. Единая система программной документации. Программа и методика испытаний. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.301-2000 ЕСПД. Введ. 01.09.2001 – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации, 2000. – 9 с.
5. Общие требования по оформлению курсовых проектов (работ), дипломных проектов, отчётов по практикам, рефератов, и других письменных работ учащихся: СТП ПГЭК 1.01 – УО «ПГЭК».
6. Системный анализ предметной области [Электронный ресурс] п–рРежимрдоступа:[аhttps://studwood.net/1785641/informatika/obsledovanie\_predmetnoy\_oblasti\_restoran](https://studwood.net/1785641/informatika/obsledovanie_predmetnoy_oblasti_restoran) – Дата доступа: 10.09.2022 г.
7. Программный комплекс R-Keeper [Электронный ресурс] – Режим доступа: <https://www.cleverence.ru/articles/elektronnaya-kommertsiya/sistema-r-keeper-chto-eto-takoe-printsip-raboty-programmy-luchshee-programmnoe-obespechenie-dlya-res/> – Дата доступа: 15.10.2022 г.
8. Программный комплекс iiko [Электронный ресурс] – Режим доступа: <https://iiko.isab.by/?yclid=3179316091705918440> – Дата доступа: 19.10.2022 г.
9. Программный комплекс Poster iiko [Электронный ресурс] –Режим доступа: <https://joinposter.com> – Дата доступа: 22.10.2022 г.
10. База данных SQLite [Электронный ресурс] – Режим доступа: [https://www.sqlite.org/index.html](https://www.sqlite.org/index.html%20) – Дата доступа: 02.11.2022 г.
11. Среда разработки Android Studio [Электронный ресурс] – Режим доступа: [https://developer.android.com/studio/install](https://developer.android.com/studio/install%20) – Дата доступа: 07.11.2022 г.

# ПРИЛОЖЕНИЕ А (обязательное) Диаграмма вариантов использования

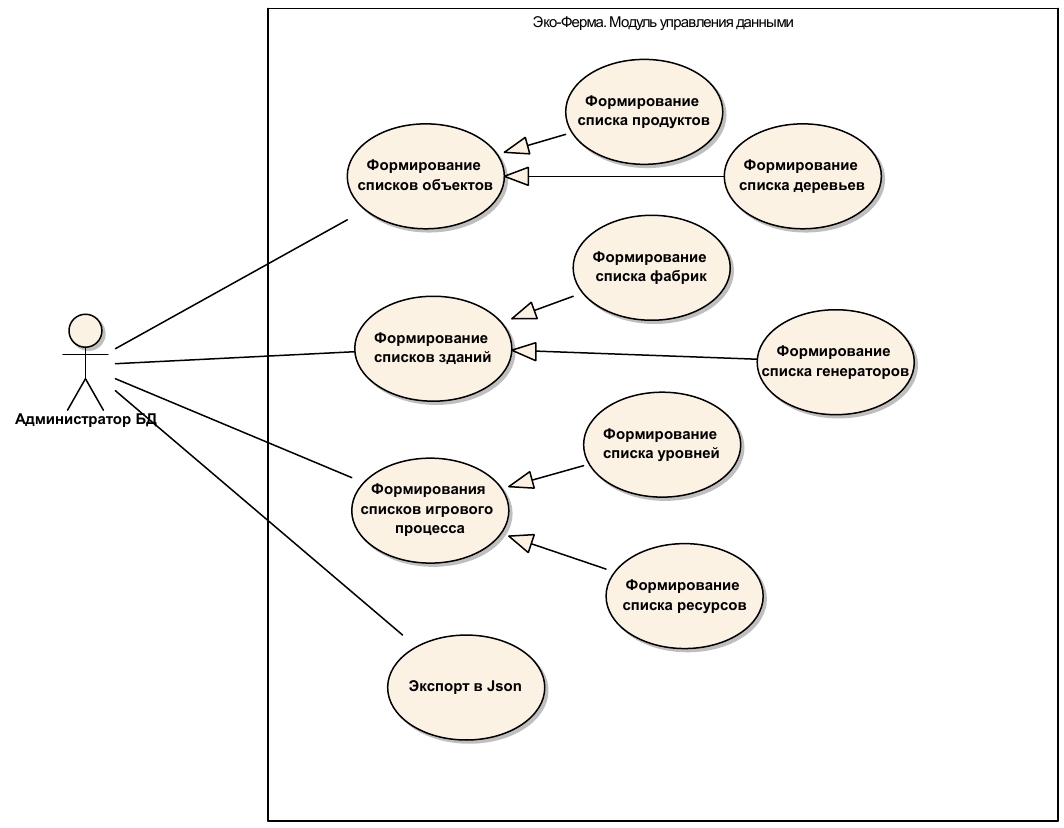


Рисунок А – Диаграмма вариантов использования

# ПРИЛОЖЕНИЕ Б (обязательное) Техническое задание

**Б Введение**

Настоящее техническое задание распространяется на разработку программы «Эко-Ферма. Модуль управления данными».

Основное назначение системы – автоматизация формирования каталога игровых сущностей и статистики игрового процесса.

**Б1 Основания для разработки**

Программа «Эко-Ферма. Модуль управления данными» разрабатывается в соответствии с заданием на курсовое проектирование по дисциплине «Базы данных и системы управления базами данных».

**Б2 Назначение разработки**

**Б2.1 Функциональное назначение**

Функциональное назначением приложения системы – автоматизация формирования каталога игровых сущностей и статистики игрового процесса. Пользователями являются: Администратор БД.

**Б2.2 Эксплуатационное назначение**

Эксплуатационное назначение системы – использование администратором базы данных игрового приложения «Эко-ферма».

**Б3.3 Требования к программе**

**Б3.1 Требования к функциональным характеристикам**

**Б3.1.1 Требования к составу обрабатываемых данных**

Программа должна обеспечить обработку следующих данных:

1. об ОР (название, описание, цена);
2. о целях (целевое количество, целевой ОР);
3. о уровнях (цели уровня, ОР на старте).

**Б3.1.2 требования к разграничению прав доступа к информации**

В системе должен быть предусмотрен один уровень доступа к информации:

* «Администратор».

**Б3.1.3 Требования к перечню функций под ролью «Администратор»**

Пользователь под ролью «Администратор» должен выполнять следующие функции:

* формирование списка продуктов;
* формирование списка деревьев;
* формирование списка генераторов;
* формирование списка фабрик;
* формирование списка уровней;
* формирование списка ресурсов;
* экспорт данных в json.

**Б3.2 Требования к надёжности**

Система должна обеспечивать следующие функции по проверки корректности ввода данных:

* проверка заполнения обязательных полей;
* вывод соответствующих информационных сообщений пользователю;

**Б3.3 Условия эксплуатации**

Система должна обеспечивать следующие условия эксплуатации:

– условие эксплуатации в соответствии с СанПин 2.2.2.542 – 96;

– предусмотреть защиту от некорректных действий пользователя;

– обеспечить целостность информации в базе данных.

**Б3.4 Требования к составу и параметрам технических средств**

Требования к составу и параметрам технических средств:

– процессор: «Intel Core 2 Duo» и новее;

– ОЗУ: 4 ГБ;

– место на жестком диске: не менее 500 МБ;

– видеоадаптер: DirectX 9 или более поздняя версия.

**Б3.5 Требования к информационной и программной совместимости**

Приложение должно быть установлено на устройство, содержащее:

* операционная система: Windows 10 или новее;
* библиотеки .NET Core 7.0 или новее;
* СУБД SQLite.

**Б3.6 Требования к маркировке и упаковке**

Специальные требования к маркировке и упаковке не предъявляются.

**Б4 Требования к программной документации**

Программная документация должна быть представлена в следующем составе:

* «Руководство оператора» – ГОСТ 19.505-79;
* «Программа и методика испытаний» – ГОСТ 19.301-2000.

**Б5 Технико-экономические показатели**

Улучшение технико-экономических показателей достигается за счёт простоты использования по сравнению с аналогичными программами.

Приложение предназначено для автоматизации формирования каталога игровых сущностей и статистики игрового процесса.

**Б6 Стадии и этапы разработки**

Разработка приложения включает в себя следующие стадии, представленные в таблице В.1.

Таблица В.1 – Стадии разработки приложения

|  |  |
| --- | --- |
| Наименование этапов курсового проектирования | Срок выполнения |
| 1 | 2 |
| 1. Описание предметной области | до 25.09.2022 |
| 1. Разработка диаграммы «Сущность-связь» | до 25.09.2022 |
| 1. Разработка диаграммы вариантов использования | до 30.09.2022 |
| 1. Разработка технического задания | до 15.10.2022 |
| 1. Разработка 1-го раздела ПЗ КП | до 20.10.2022 |
| 1. Проектирование интерфейса программного средства | до 23.10.2022 |
| 1. Обоснование выбора инструментальных средств | до 23.10.2022 |
| 1. Разработка физической модели данных | до 30.10.2022 |
| 1. Разработка 2-го раздела ПЗ КП | до 10.11.2022 |
| 1. Реализация программного средства | до 25.11.2022 |
| 1. Разработка программного документа «Программа и методика испытаний», тестирование ПО | до 30.11.2022 |
| 1. Разработка эксплуатационной документации | до 30.11.2022 |
| 1. Разработка 3-го раздела ПЗ КП | до 10.12.2022 |
| 1. Оформление курсового проекта (ПЗ, чертежи, электронный носитель) | до 14.12.2022 |

**Б7 Порядок контроля и приёмки**

**Б7.1 Порядок контроля**

Контроль выполнения осуществляется преподавателем дисциплины БДиСУБД.

**Б7.2 Порядок защиты**

Защита осуществляется перед руководителем практики.

**Б7.3 Срок защиты**

Срок защиты определяется расписанием занятий в соответствии с графиком учебного процесса.

# ПРИЛОЖЕНИЕ В (обязательное) Диаграмма сущность-связь

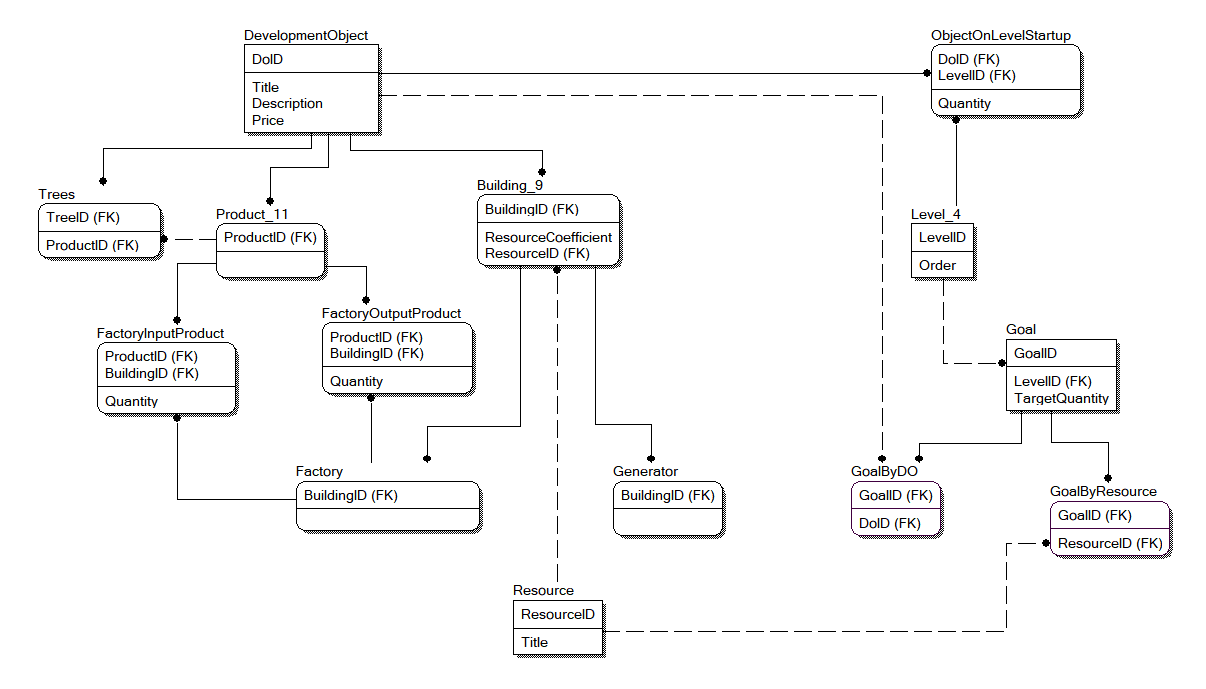


Рисунок В – Диаграмма

# ПРИЛОЖЕНИЕ Г (справочное) Диаграмма развертывания



Рисунок Г.1 – Диаграмма развёртывания

# ПРИЛОЖЕНИЕ Г (обязательное) Руководство оператора

**Г1 Назначение программы**

Программа «Эко-Ферма. Модуль управления данными» разработана в соответствии с индивидуальным заданием.

**Г2 Условие выполнения программы**

Для функционирования программы, необходимо следующие аппаратное обеспечение:

– процессор: «Intel Core 2 Duo» и новее;

– ОЗУ: 4 ГБ для 32-разрядной системы и 4 ГБ для 64-разрядной системы;

– место на жестком диске: не менее 500 МБ;

– видеоадаптер: DirectX 9 или более поздняя версия.

Также для работы с программным средством, администраторы должны обладать знаниями работы с персональным компьютером на уровне оператора ЭВМ.

**Г3 Выполнение программы**

Для запуска программы требуется выполнить двойной клик левой кнопкой мыши по исполняемому файлу «EcoFarmAdmin.exe».

**Г3.1 Функционал пользователя «Администратор»**

После запуска программы пользователь попадет на окно «Меню», которая представлена на рисунке Г1.

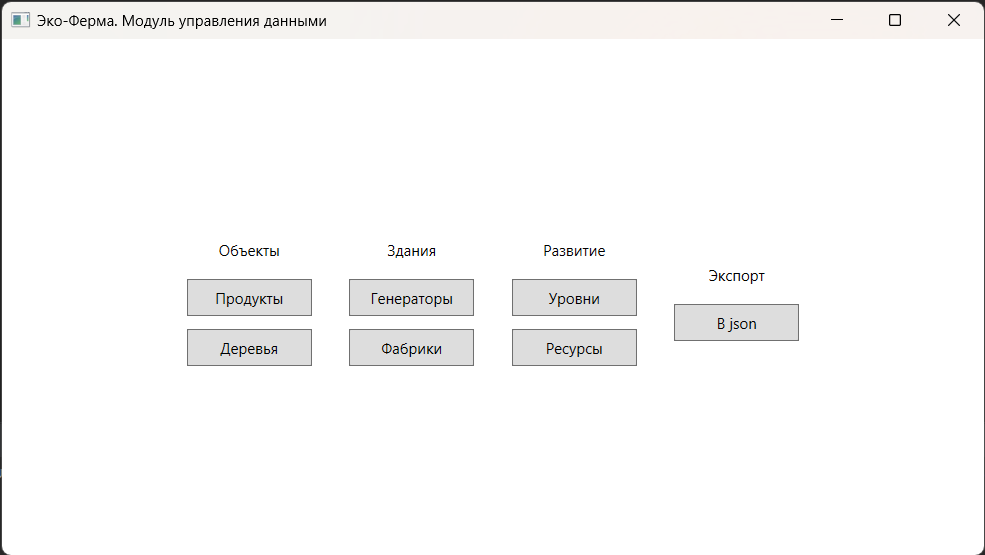


Рисунок Г1 – Основное окно

На окне присутствуют 7 кнопок: «Продукты», «Деревья», «Генераторы», «Фабрики», «Уровни», «Ресурсы» и «Экспорт в json».

**Г3.2 Окно «Продукты»**

Окно «Продукты» представлено на рисунке Г2.

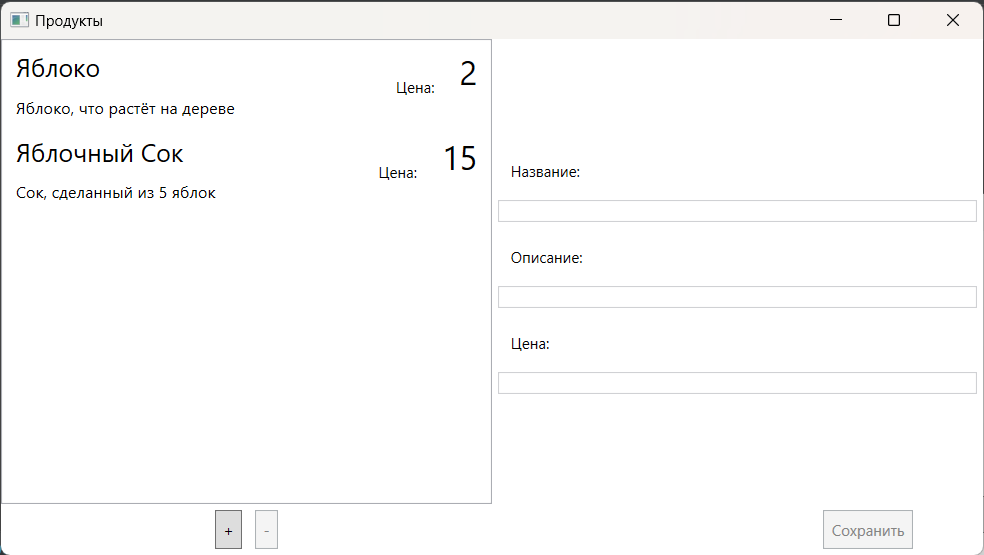


Рисунок Г2 – Окно «Продукты»

Для добавления продукта, нужно нажать на кнопку «+» внизу экрана, в список добавится пустая запись, которую можно будет заполнить используя поля справа (Рисунок Г3). Редактирование происходит аналгично, за исключением того, что пользователю необходимо выбрать уже существующую запись.

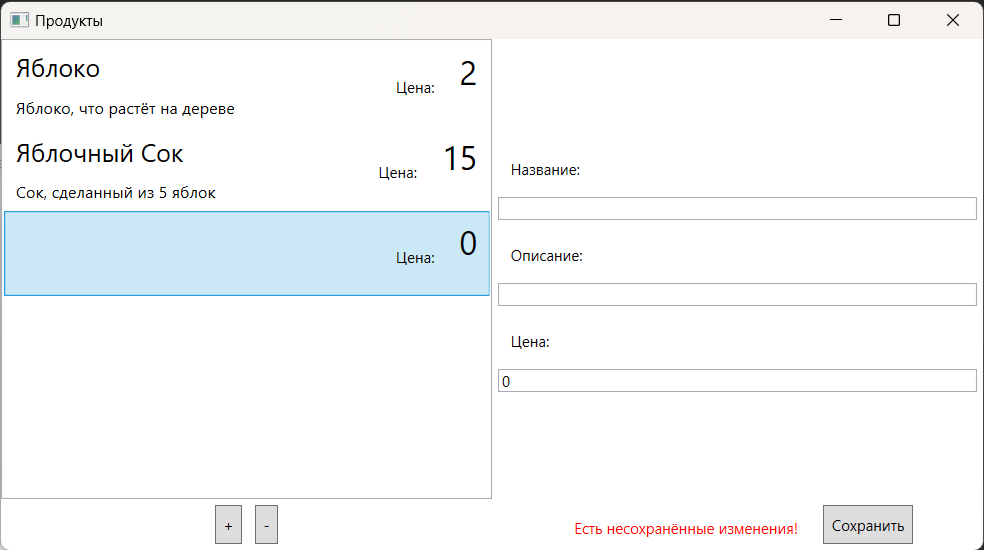


Рисунок Г3 – Добавление продукта

Для того чтобы удалить запись, нужно выделить целевой продукт и нажать по кнопке «-».

Окно подтверждения удаления представлено на рисунке Г4.

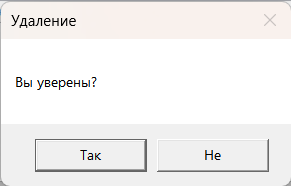


Рисунок Г4 – Окно подтверждения удаления

нужно ли расписывать аналогично другие окна?

**Г3.3 Окно «Уровни»**

Окно «Уровни» представлена на рисунке Г.5.

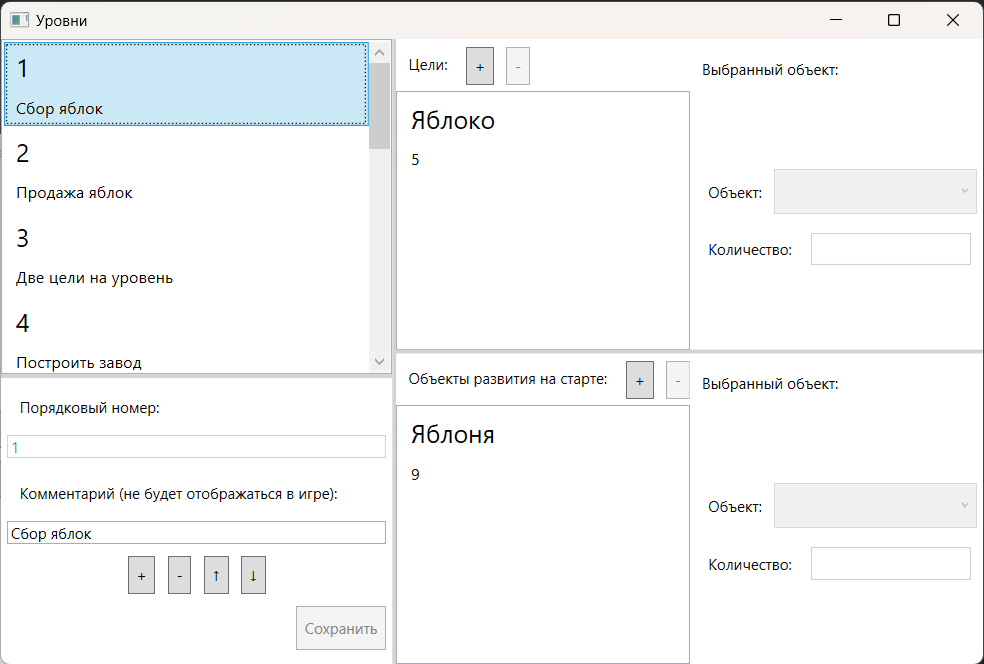


Рисунок Г5 – Окно «Уровни»

Для добавления уровня, нужно нажать на кнопку «+» внизу экрана, в список добавится пустая запись, которую можно будет заполнить (Рисунок Г.6). Редактирование происходит аналогично, за исключением того, что пользователю необходимо выбрать уже существующую запись.

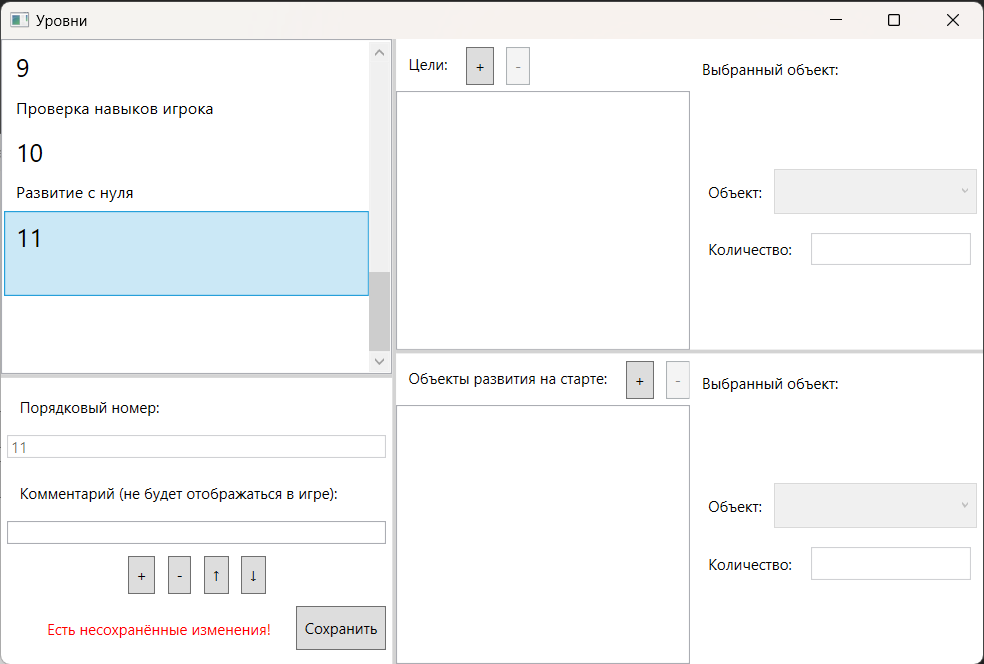


Рисунок Г6 – Добавление нового уровня

На каждый уровень пользователь может добавлять цели и объекты на старте. Для этого ему необходимо выбрать нужный уровень и нажать кнопку «+» в соответствующем поле. После чего он сможет выбрать ОР из выпадающего списка и задать целевое количество объектов (Рисунок Г9). Редактирование и удаление происходит аналогично, как и для остальных таблиц.

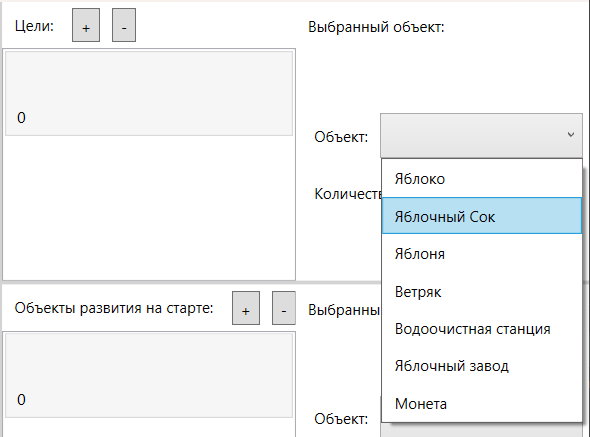


Рисунок Г7 – Добавление цели уровня

Если пользователь добавил цель или ОР на старте, но не выбрал объект из выпадающего списка – то при попытке сохранения система выдаст ему сообщение об ошибке.

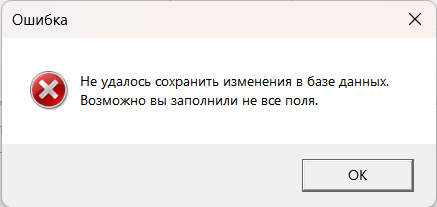


Рисунок Г8 – Сообщение об ошибке при сохранении

**Г3.4 Экспорт в json**

Для экспортирования БД в json файл, необходимо нажать на соответствующую кнопку на главном окне (Рисунок Г1). Тогда пользователю откроется диалоговое окно выбора дирректории для сохранения файла

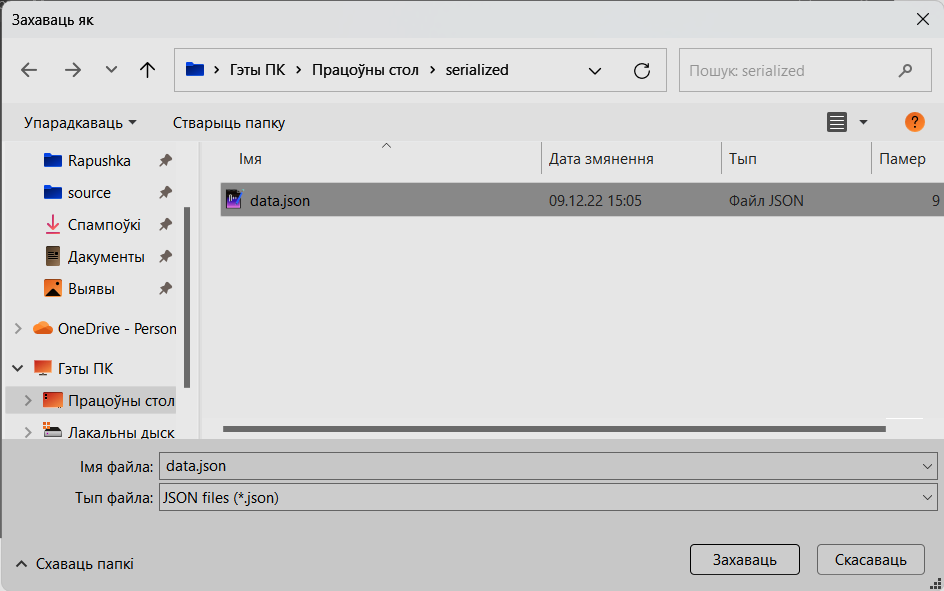


Рисунок Г9 – Выбор целевой дирректории

По завершении экспорта, система покажет соответствующее сообщение с полным путём к папке, куда был сохранён файл (Рисунок Г10).

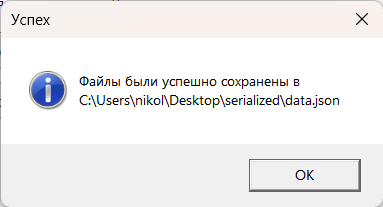


Рисунок Г10 – Сообщение о успешном экспорте

После чего, в указаной пользователем дирректории, появится файл, содержащий сериализование данные в формате json.

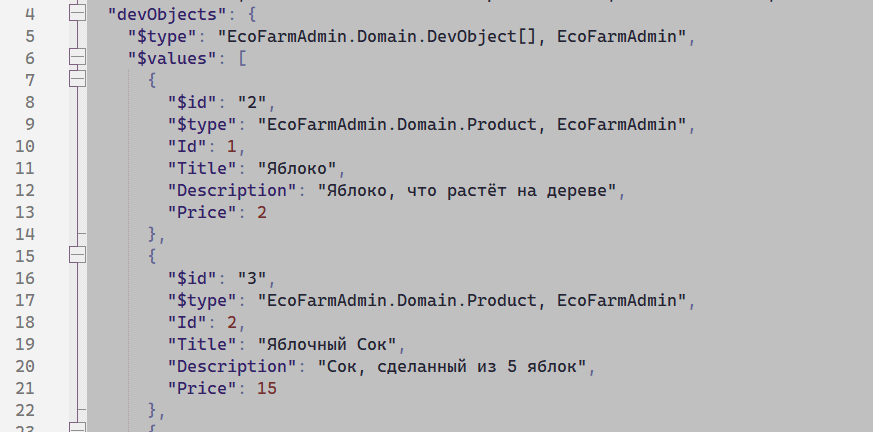


Рисунок Г11 – Часть сериализованых данных

**Г3.5 Интеграция с Unity**

Данный json файл подготовлен специально для использования в Unity проекте EcoFarm. Где он должен быть выбран, для десереализации данных о игровых сущностях.

В проекте эти данные используются в двух сценариях: непосредственно в геймплее (Рисунок Г12) и в редакторе, для выбора спрайтов для каждой игровой сущности (Рисунок Г13).



Рисунок Г12 – Уровень сгенерированный по данным из файла

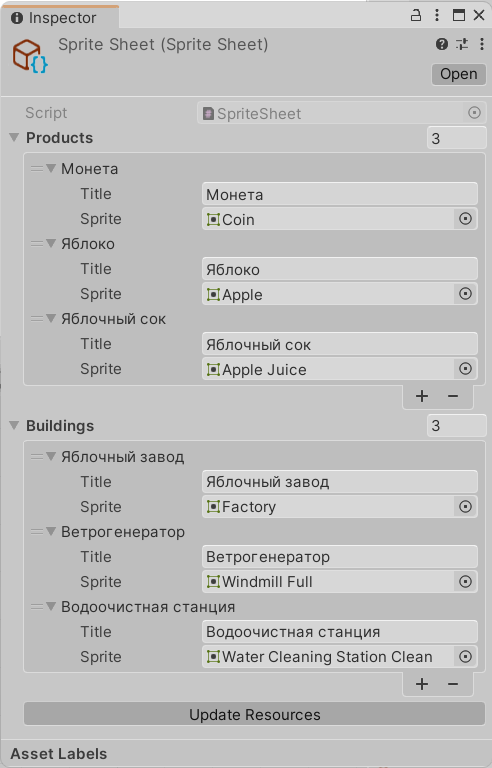


Рисунок Г13 – Окно установки спрайтов игровым объектам в Unity

**Г4 Сообщения оператору**

Программа выдаёт ошибки, а также предупреждение во время некорректного ввода данных, тексты с ошибками и их решениями представлены в таблице Г.4.1.

Таблица Г4.1 – Сообщения оператору

|  |  |  |
| --- | --- | --- |
| Текст сообщения | Причина появления | Требуемые действия |
| Не удалось сохранить изменения в базе данных. Возможно вы заполнили не все поля. | Возникла ошибка при попытке сохранения внесённых изменений в базу данных | Проверить, не осталось ли пустых значений в выпадающих списках. |

# ПРИЛОЖЕНИЕ Д (Обязательное) Программа и методика испытаний

**Д1 Объект испытаний**

Объектом испытания является программа «Эко-ферма. Модуль управления данными».

Пользователями системы могут быть администратор БД.

**Д2 Цель испытаний**

Целью проведения испытаний является установление работоспособности разработанной информационной системы в соответствии техническому заданию и другим программным документам. При выявлении несоответствий работы приложения техническому заданию, либо ошибок в программной части, требуется доработка. Тестирование программы должно:

* подтвердить работоспособность системы в соответствии с техническим заданием на разработку;
* выявить все ошибки, исключить недоработки и несоответствия техническому заданию.

Таким образом, главной целью испытаний является выявление как логических, так и функциональных ошибок программы.

**Д3 Требование к программе**

В системе должны быть реализованы уровни доступа к информации под следующими ролями: Администратор, Система.

Количество потенциально-возможных пользователей информационной системы в один момент времени – 1.

* предусмотреть контроль вводимой информации;
* предусмотреть защиту от некорректных действий пользователя;
* обеспечить целостность информации в локальной системе хранения данных.

**Д4 Требования к программной документации**

Разработанная конфигурация должна быть укомплектована следующими программными документами:

* «Программа и методика испытаний»;
* «Руководство оператора».

Программная документация должны быть оформлена в соответствии со следующими нормативными документами:

* «Руководство оператора» – ГОСТ 19.505-79;
* «Программа и методика испытаний» – ГОСТ 19.301-2000.

**Д5 Средства и порядок испытаний**

Программа должна функционировать на персональных компьютерах с минимальной конфигурацией технических средств для клиентского приложения:

Требования к составу и параметрам технических средств:

– процессор: «Intel Core 2 Duo» и новее;

– ОЗУ: 4 ГБ для 32-разрядной системы и 4 ГБ для 64-разрядной системы;

– место на жестком диске: не менее 500 МБ;

– видеоадаптер: DirectX 9 или более поздняя версия.

Порядок испытаний:

1. Проверка формирования каталога Объектов развития:

* импорт;
* добавление;
* удаление;
* редактирование.

1. Формирование каталога уровней:

* импорт;
* добавление;
* удаление;
* редактирование.

1. Формирование каталога целей:

* импорт;
* добавление;
* удаление;
* редактирование.

1. Формирование каталога зданий:

* импорт;
* добавление;
* удаление;
* редактирование.

1. Импорт базы данных.
2. Экспорт базы данных.

9) Сериализация файлов.

**Д6 Методы испытаний**

Тестирование программного продукта должно проводится на исправном оборудовании, с набором программных средств, заявленными в настоящем документе и техническом задании. При тестировании сверяется ожидаемый и фактический результат, если результаты совпадают, то испытание считается успешным, если возникают ошибки, то испытание считается неуспешным. Для данного программного модуля проводится функциональное тестирование методом «черного» ящика для которого используется тест-план, представленный в таблице Д.1. Сценарий тестирования ссылается на пункты в «Руководство оператора» в ПРИЛОЖЕНИИ Г.

Таблица Д.1 – Тест-план тестирования ПО

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | № тест требования | Описание теста | Входные данные | Ожидаемый результат | Сценарий тестирования |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1.1 | Авторизация Администратора |  | Открытая главная форма | Пункт Ж1 |
| 2 | 2.1 | Добавление записи | Порядковый номер – 1 | Запись добавлена | Пункт Ж4 |
| 3 | 2.1 | Добавление некорректной записи | Порядковый номер – АБВ | Сообщение об ошибке | Пункт Ж5 |
| 4 | 2.2 | Редактирование записи | Order – 10 | Запись изменена | Пункт Ж6 |
| 5 | 2.3 | Удаление записи | Запись с номером – 1 | Запись удалена | Пункт Ж7 |
| 6 | 2.3 | Попытка удаления записей без выбора |  | Кнопка удаления не доступна | Пункт Ж8 |
| 7 | 3.1 | Сохранение БД |  | Данные занесены в файл БД | Пункт Ж9 |
| 8 | 3.1 | Попытка сохранить неоткрытую БД |  | Сообщение об ошибке | Пункт Ж10 |
| 9 | 4.2 | Сериализация | Раположение: рабочий стол | Сериализованный файл | Пункт Ж12 |

# ПРИЛОЖЕНИЕ Е (обязательное) Результаты тестирования

Таблица – Результаты тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тестовый случай | Входные данные | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | 2 | 3 | 4 | 5 |
| Авторизация Администратора |  | Открытая главная форма | Открытая главная форма | Тест пройден |
| Создание Нового файла БД | Расположение: рабочий стол | Открытый файл БД | Открытый файл БД | Тест пройден |
| Импорт файла БД | Расположение: рабочий стол | Открытый файл БД | Открытый файл БД | Тест пройден |
| Добавление записи | Id – 0;  Order – 1 | Запись добавлена | Запись добавлена | Тест пройден |
| Добавление некорректной записи | Id – АБВ | Сообщение об ошибке | Сообщение об ошибке | Тест пройден |
| Редактирование записи | Order – 10 | Запись изменена | Запись изменена | Тест пройден |
| Удаление записи | Запись с Id – 1 | Запись удалена | Запись удалена | Тест пройден |
| Попытка удаления записей без выбора |  | Сообщение об ошибке | Сообщение об ошибке | Тест пройден |
| Сохранение БД |  | Данные занесены в файл БД | Данные занесены в файл БД | Тест пройден |
| Попытка сохранить неоткрытую БД |  | Сообщение об ошибке | Сообщение об ошибке | Тест пройден |
| Десериализация файла | Раположение: рабочий стол | Данные подгрузились | Данные подгрузились | Тест пройден |

# ПРИЛОЖЕНИЕ Ж (обязательное) Содержание электронного носителя

На прилагаемом электронном носителе информации расположены следующие директории и файлы:

1. Директория «DOC» содержит файл «П42\_Сятковский\_Н.В.\_П\_ЭкоФерма\_Модуль\_управления\_данными.docx» и представляет собой документ с пояснительной запиской.
2. Директория «PRG» содержит исходный код проекта «EcoFarmAdmin.csproj».
3. Директория «EXE» содержит финальную версию исполняемого файла проекта «EcoFarmAdmin.exe» и файл базы данных «EcoFarm.db».
4. Файл «Readme.txt» содержит описание содержания электронного носителя.