

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

*Факультет компьютерных наук  
Кафедра программирования и информационных технологий*

*Разработка приложения «Deer Build Galactic»*

*Курсовая работа*

*09.03.04 Программная инженерия*

Студенты \_\_\_\_\_ *Евтеев Е.А.*  
*подпись*

Преподаватель \_\_\_\_\_ *Тарасов В.С.*  
*подпись*

Преподаватель \_\_\_\_\_ *Клейменов И.В.*  
*подпись*

Зав. Кафедрой \_\_\_\_\_ *Махортов С.Д.*  
*подпись*

Воронеж 2022

## Содержание

1. Анализ предметной области .....	4
1.1. Постановка задачи .....	4
1.2. Обзор аналогов .....	5
1.2.1. PoE Planner – <a href="http://poeplanner.com">poeplanner.com</a> .....	5
1.2.2. Overframe – <a href="http://overframe.gg">overframe.gg</a> .....	7
1.3. Терминология и сокращения .....	9
1.4. Функциональные требования .....	9
1.5. Технические требования .....	10
1.6. Требования к интерфейсу .....	10
2. Разработка .....	11
2.1. Макеты интерфейса .....	11
2.2. Диаграмма IDEF-0 .....	11
2.3. Диаграмма Use-case .....	12
2.4. Диаграмма состояний .....	12
2.5. Диаграмма активностей .....	13
2.6. Диаграмма DFD .....	14
2.7. Схема Базы данных .....	14
2.8. Технические средства .....	15
2.9. Состав и содержание работ по созданию системы .....	15
3. Реализация .....	18
3.1. MainForm .....	18

3.2. API .....	18
3.3. Дизайн .....	18
4. Тестирование .....	18
4.1. Результаты тестирование .....	21
5. Список использованной литературы.....	22
6. Заключение .....	22

## **1. Анализ предметной области**

Deer Rock Galactic – это совместная видеоигра -шутер от первого лица для 1-4 игроков, в которой игроки работают вместе, исследуя разрушаемые процедурно генерируемые пещерные системы. Игра ориентирована на бои PvE.

Deer Rock Galactic предоставляет своим игрокам на выбор огромный арсенал Снаряжения, предназначенного для помощи гномам в добыче ресурсов и устранении угроз. Все классы несут сразу несколько предметов снаряжения – основное оружие, дополнительное оружие, два вспомогательных инструмента и метательный предмет (например, гранату).

Что касается оружия, у гнома есть три варианта слотов основного и дополнительного оружия. Альтернативные варианты слота для оружия должны быть разблокированы с помощью лицензии на оборудование, а затем покупки указанного оружия. Инструменты поддержки помогают гномам с мобильностью и полезностью – у каждого класса есть уникальная пара инструментов поддержки, связанных с их работой. Наконец, класс может выбрать один из трех различных вариантов метательного, хотя последние два варианта должны быть разблокированы и куплены в первую очередь.

Наряду с этими предметами у каждого гнома есть один и тот же базовый набор инструментов – кирка для добычи ресурсов и нанесения урона в ближнем бою, броня, лазерная указка, чтобы отмечать точки, представляющие интерес для команды, сканер местности, который действует как трехмерная карта, и метательные сигнальные ракеты, которые служат временными источниками света.

### **1.1. Постановка задачи**

Целью данной работы является разработка приложения способного упростить для пользователя, создание наборов снаряжения для своего персонажа. Разработанное приложение должно обладать интуитивно понятным интерфейсом.

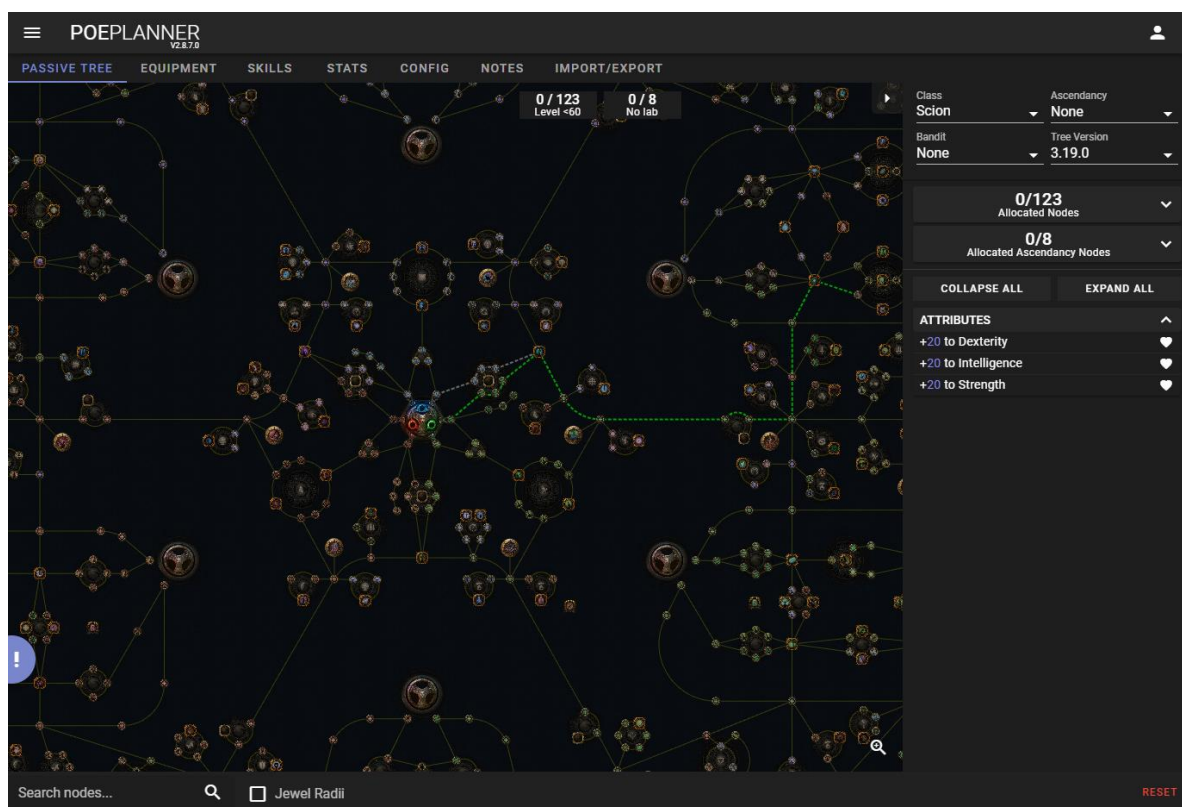
Разрабатываемый проект должен удовлетворять следующим основным требованиям и реализовывать возможности:

- Создавать билд экипировки для класса;
- Удалять билд;
- Редактировать билд;
- Сохранять билд в файле;
- Загружать билд из файла.

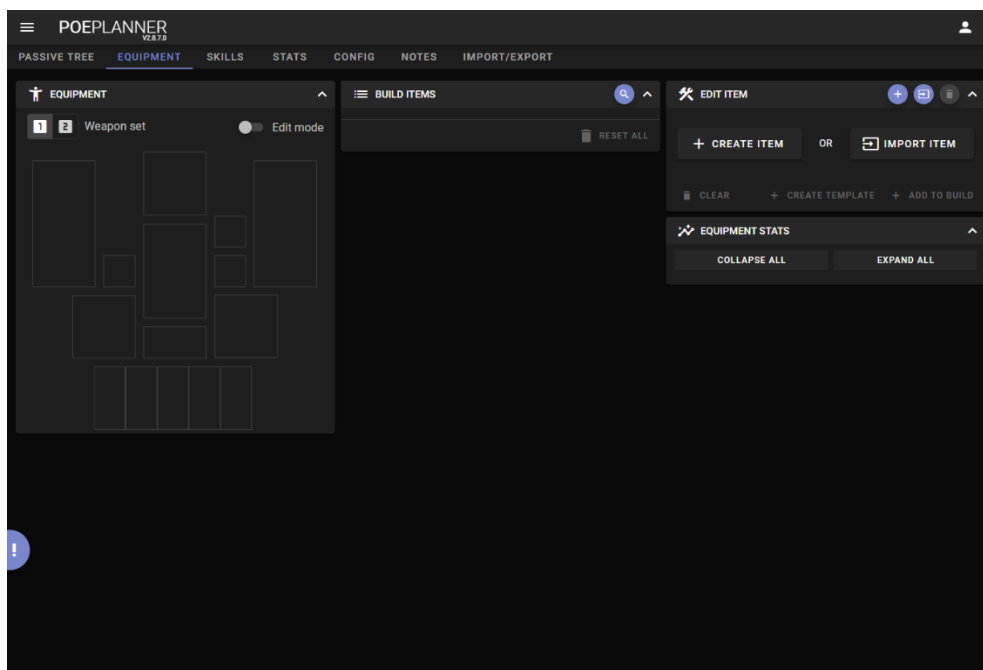
## 1.2. Обзор аналогов

При инициализации разработки необходимо произвести анализ ближайших аналогов имеющие схожий функционал.

### 1.2.1. PoE Planner – poeplanner.com



Главная страница сервиса



## Страница снаряжения

### Функциональность:

- Поиск наборов снаряжения.
- Создание снаряжение.
- Просмотр модов.
- Настройка сервиса.
- Калькулятор характеристик.

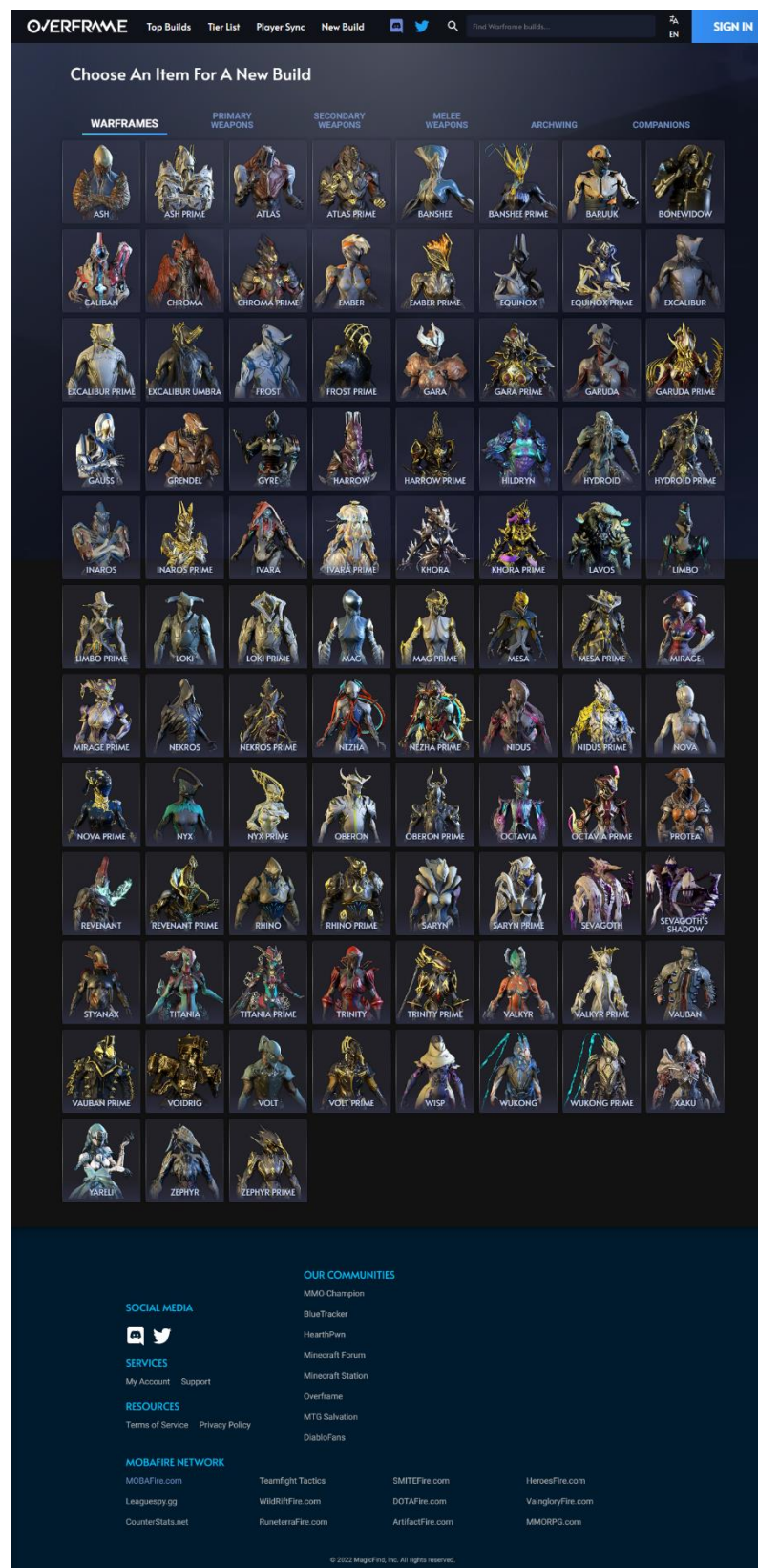
### Достоинства:

- Уникальный функционал по созданию снаряжения (согласно документации разработчиков).
- Поддержка со стороны разработчиков.
- Импорт и Экспорт.

### Недостатки:

- Отсутствие локализации.

## 1.2.2. Overframe – overframe.gg



Главная страница





#### Функциональность:

- Поиск модов названию или фильтрам.
- Сортировка модов странице.
- Обзор пользовательских наборов.
- Рейтинг наборов снаряжений.

#### Достоинства:

- Синхронизация с игрой.
- Создание тир-листов.

#### Недостатки:

- Отсутствие локализации.

### 1.3. Терминология и сокращения

- Проект – разрабатываемое приложение;
- C# (читается Си-Шарп) – высокоуровневый объектно-ориентированный язык программирования;
- GitHub – веб-сервис для хостинга и совместной разработки IT-проектов;
- Билд – набор улучшений;
- Оверлок – «модуль разгона», улучшение.
- Пользователь – человек, имеющий доступ к базовому функционалу приложения.

### 1.4. Функциональные требования

Система должна соответствовать следующим функциональным требованиям:

1. Создание нового билда

- При запуске программы пользователю будет предложено создать билд или открыть существующий из файла.

## 2. Просмотр билда

- Пользователь может просматривать составляющие билда.

## 3. Сохранение билда

- Созданный пользователем билд можно сохранить в виде файла.

# 1.5. Технические требования

Созданное приложение должно иметь архитектуру десктопного приложения. Операционная система семейства ОС Windows начиная с Windows версии 7, так как это последняя версия официально поддерживаемая Microsoft .NET 6.

# 1.6. Требования к интерфейсу

Оформление должно соответствовать следующим требованиям:

- Все экраны должны соответствовать единому дизайну.
- Вёрстка должна быть адаптивной (реализовывать оконное расширение).
- Все элементы управления должны быть выделены на фоне основного содержимого экрана.

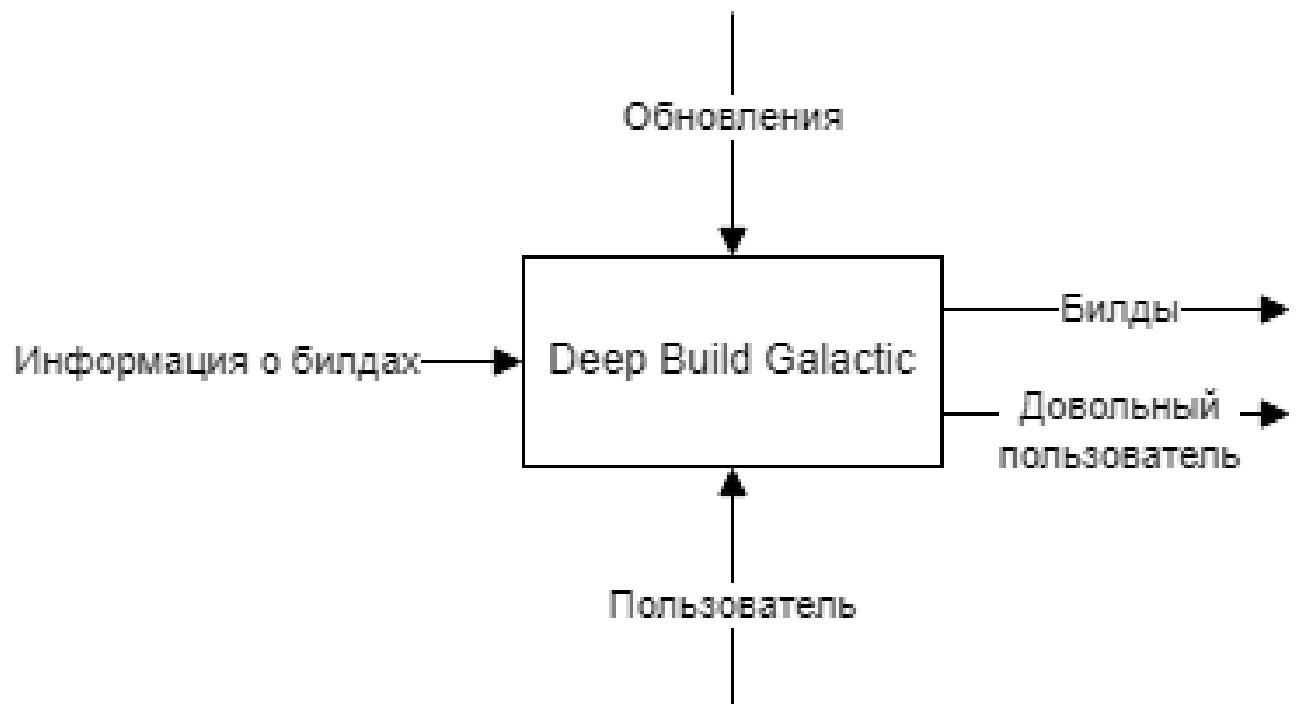
## 2. Разработка

### 2.1. Макеты интерфейса

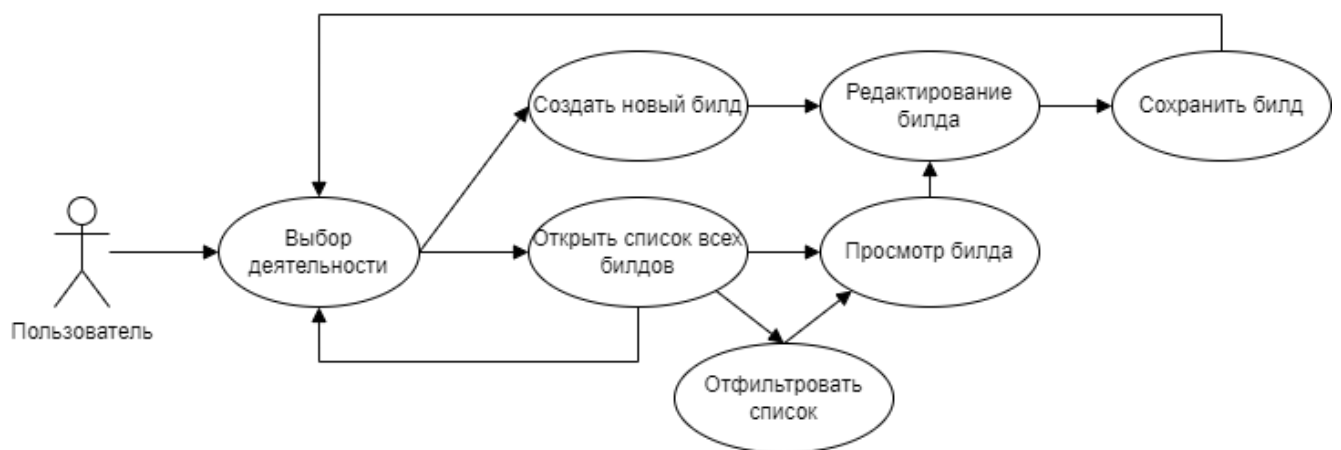


Экран создания билда

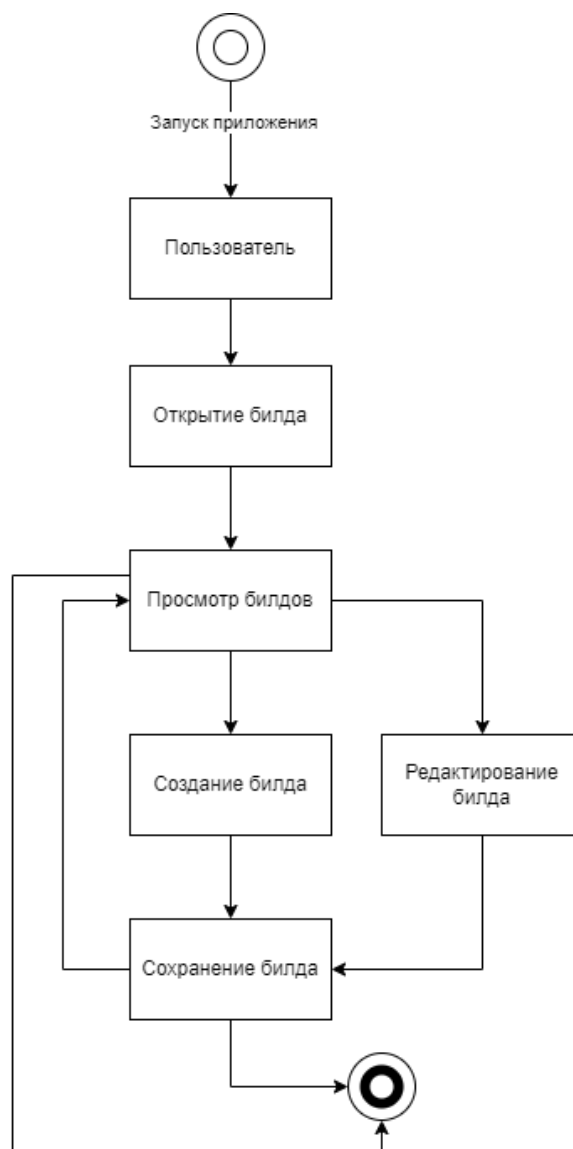
### 2.2. Диаграмма IDEF-0



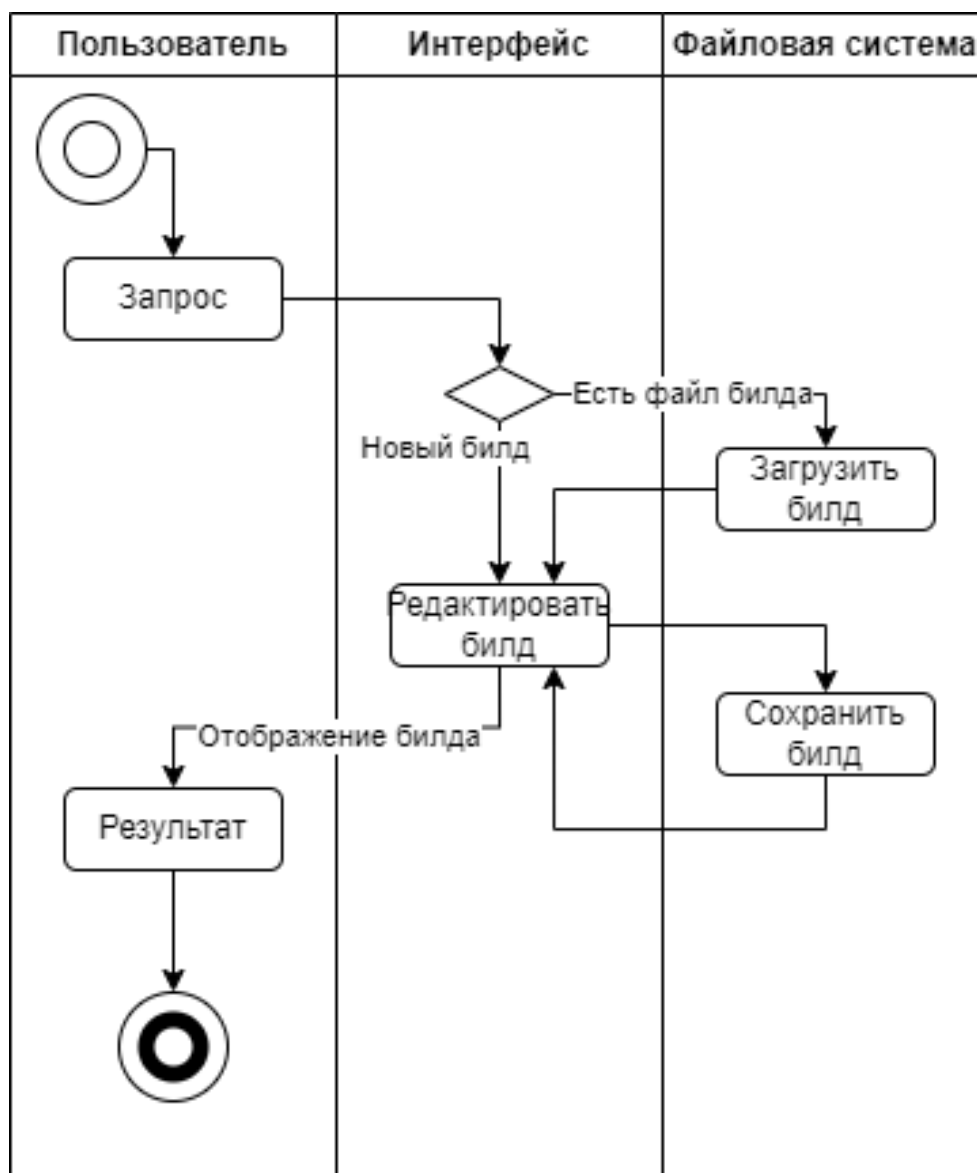
### 2.3. Диаграмма Use-case



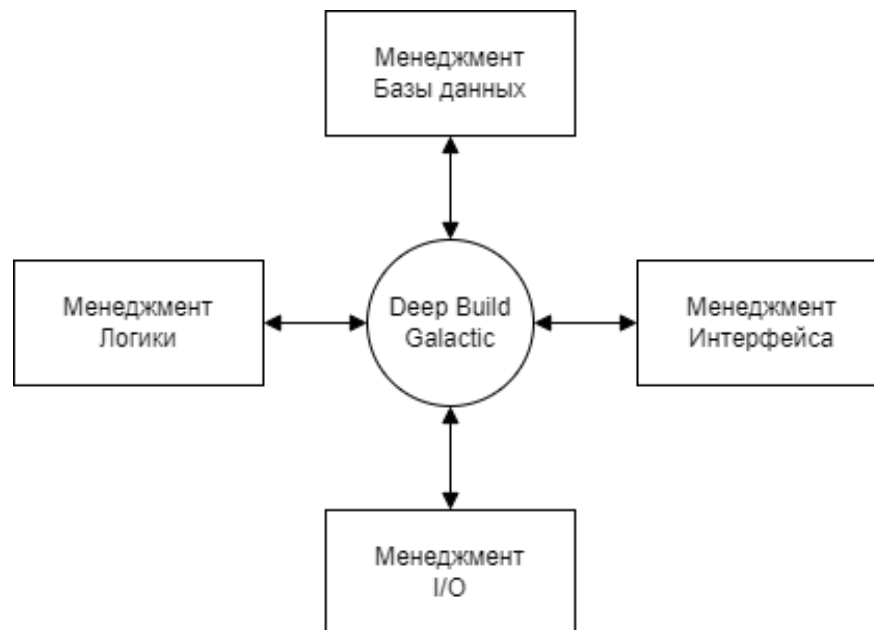
### 2.4. Диаграмма состояний



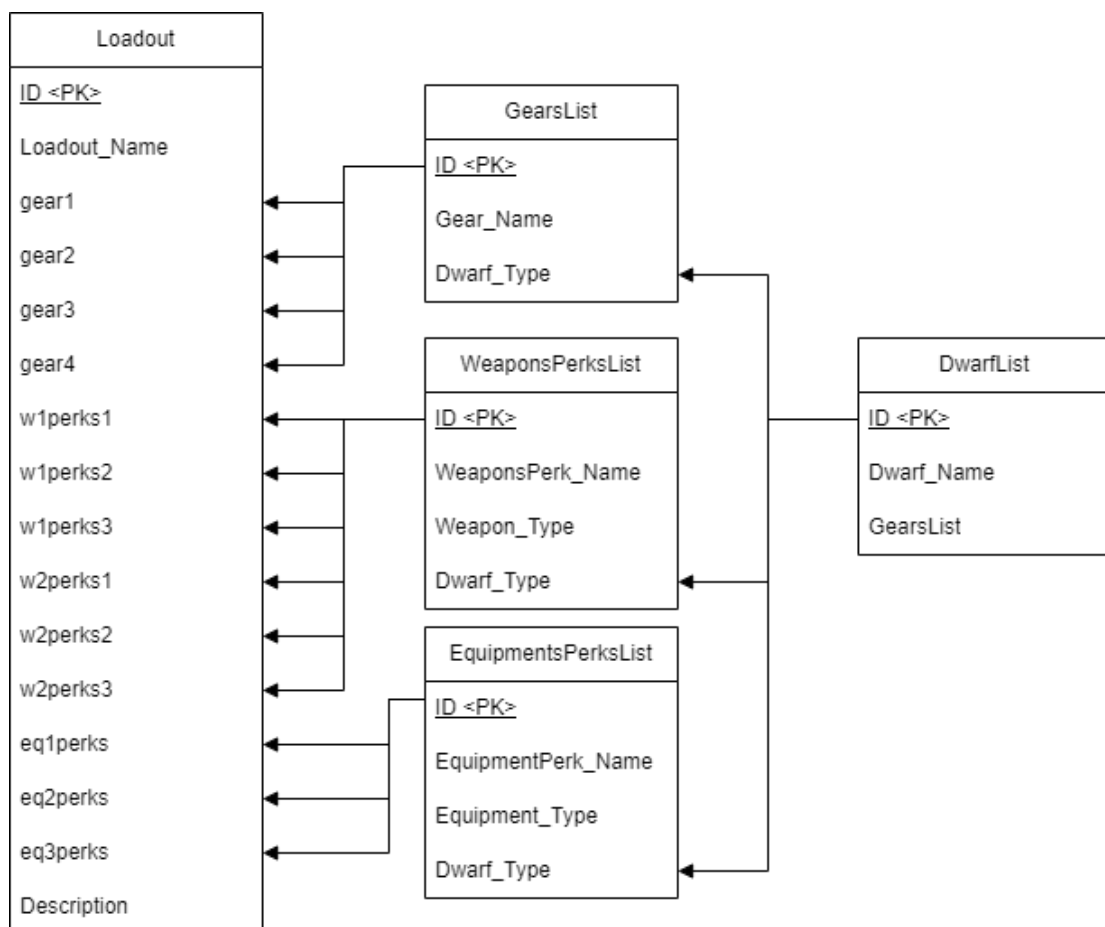
## 2.5. Диаграмма активностей



## 2.6. Диаграмма DFD



## 2.7. Схема Базы данных



## 2.8. Технические средства

- Язык программирования C#
- Базовые библиотеки работы с графикой C# (WPF)
- IDE Visual Studio 2022
- Система контроля версий Git (Github)
- Task-менеджер Github Projects
- Paint.NET

## 2.9. Состав и содержание работ по созданию системы

Плановый срок начала работ – Сентябрь 2022 года

Плановый срок окончания работ – Ноябрь 2022 года

Основные этапы по разработке системы, их содержание и примерные сроки выполнения приведены в Таблице 1

Этап	Содержание работ	Порядок приёмки документы	Сроки	Ответственный
1. Техническое задание	Разработка требований к системе	Утверждение технического задания	До 16.09.22	Разработка – Исполнитель Согласование – Заказчик
2. Техническое проектирование	Разработка сценариев работы системы	Предоставление сценариев	До 16.09.22	Исполнитель

	Разработка дизайн-макета	Представлени е изображений дизайн-макета	До 16.09.22	Исполнитель
3. Разработка системы	Разработка логического модуля, модуля базы данных, модуля интерфейса	Приём результатов работы осуществляе тс я в процессе тестирования и испытаний	В течение 1 календарного месяца с момента утверждения технического задания	Исполнитель
4. Предваритель ные автономные испытания	Проверка соответствия функциональн ым требованиям	Согласно техническому заданию	В течение недели (7 дней) с момента завершения разработки	Исполнитель
	Проверка документации			
	Доработка и повторные испытания до устранения обнаруженны х недостатков			



5. Разработка курсового проекта и сопровождающей документации	Разработка курсового проекта на основе технического задания	В течение всего времени работы над проектом с момента утверждения технического задания	До 16.11.22	Исполнитель
6. Опытная эксплуатация	Предоставление доступа к текущей версии проекта, ограниченном у количеству участников с целью тестирования на соответствие функциональным требованиям	Создание и ведение внутреннего документа	До 16.11.22	Исполнитель

	Доработка и повторные испытания до устранения обнаруженных недостатков			
--	--	--	--	--

Таблица 1 - Основные этапы разработки системы

### 3. Реализация

#### 3.1. MainForm

Отвечает за внешний вид и работу с пользователем, а именно:

- Работа с пользователем – обработка запросов;
- Отслеживание запросов (функция адресаций–контроллеров обращается к остальным компонентам приложения);

#### 3.2. API

API отвечает за основной функционал приложения. Предоставляет пользователю работать с приложением с минимальными затратами по времени, в виду десктопной архитектуры. Благодаря API доступны возможности для расширения возможностей и дальнейшей поддержки.

#### 3.3. Дизайн

Написан на с помощью возможностей WPF (Windows Presentation Foundation) и библиотек языка C#.

### 4. Тестирование

В ходе завершающих работ, было произведено тестирование, которое покрывает основной функционал приложения.

Тест-кейсы для тестирования выглядят следующим образом.

Тест-кейс №1		
Заголовок		Проверка сохранения
Предусловия		Отсутствуют ранее созданные файлы
Доп. данные		
Шаг		Результат
1	Пользователь открывает приложение	Отображается главный экран снаряжения
2	Пользователь создаёт билд	Отображается выбранное пользователем снаряжение
3	Пользователь пытается сохранить билд	Открывается новое окно стандартного Windows «Сохранить как»
4	Пользователь вводит валидное имя для файла «mybuild»	Файл сохраняется

Таблица 2 - Тест-кейс №1

Тест-кейс №2	
Заголовок	Проверка загрузки
Предусловия	В системе есть файл билда
Доп. данные	mybuild.dbg
Шаг	Результат

1	Пользователь открывает приложение	Отображается главный экран снаряжения
2	Пользователь загружает билд	Открывается новое окно стандартного Windows «Открыть...»
3	Пользователь пытается загрузить билд	Отображается выбранное пользователем снаряжение из файла

Таблица 3 - Тест-кейс №2

Тест-кейс №3		
Заголовок		Ввод в название билда спецсимволов
Предусловия		
Шаг		Результат
1	Пользователь открывает приложение	Отображается главный экран снаряжения
2	Пользователь создаёт билд	Отображается выбранное пользователем снаряжение
3	Пользователь вводит имя билда как «@!#»	Символы вводятся в поле имени
4	Пользователь пытается сохранить билд	Программа предупреждает о наличии не валидных символов в поле «Имя билда»

Таблица 4 - Тест-кейс №3

#### 4.1. Результаты тестирования

Тестирование проекта Deep Build Galactic	
Тестирующий	Соколов В.А.
Дата тестирования	14.11.22
Тест-кейс №	1
Шаг	Результат ОК / NOK (+ примечания)
1	ОК
2	ОК
3	ОК
4	ОК

Таблица 7 – Результат тестирования 1

Тестирование проекта Deep Build Galactic	
Тестирующий	Соколов В.А.
Дата тестирования	14.11.22
Тест-кейс №	2
Шаг	Результат ОК / NOK (+ примечания)
1	ОК
2	ОК
3	ОК

Таблица 7 – Результат тестирования 1

Тестирование проекта Deep Build Galactic	
Тестирующий	Соколов В.А.
Дата тестирования	14.11.22
Тест-кейс №	3
Шаг	Результат ОК / NOK (+ примечания)
1	ОК
2	ОК
3	ОК
4	ОК

Таблица 7 – Результат тестирования 1

## 5. Список использованной литературы

1. Разработка требований к программному обеспечению. [Текст] Вигерс Карл, Битти Джой В41 3-е изд., дополненное / Пер. с англ. — М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2014. — 736 стр.: ил.
2. Документация по C# – [Электронный ресурс] – <https://learn.microsoft.com/ru-ru/dotnet/csharp/>
8. Git – [Электронный ресурс] – <https://git-scm.com/doc>

## 6. Заключение

В ходе выполнения данной курсовой работы было разработано приложение, которое отвечает, поставленным функциональным требованиям. Были

использованы современные технологии и методологии разработки десктопных приложений. При помощи выбранного стека технологий были реализованы основные модули приложения для его корректной работы.