****

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  **информационных систем**  **и технологий** | **Кафедра**  **информационных систем** |

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «**Проектирование информационных систем**»

на тему: «**Проектирование подсистемы для планирования индивидуальных тренировок по лёгкой атлетике**»

Направление **09.03.02 Информационные системы и технологии**

**Руководитель,**

ст. преподаватель **Овчинников П.Е.**

«\_\_\_» \_\_\_\_\_\_\_\_ 2018 г.

**Студент,**

группа ИДБ–15-14 **Балаганский А.А.**

«\_\_\_» \_\_\_\_\_\_\_\_ 2018 г.

Москва 2018 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc531955188)

[ГЛАВА 1. ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ (IDEF0) 4](#_Toc531955189)

[ГЛАВА 2. МОДЕЛЬ ПОТОКОВ ДАННЫХ (DFD) 8](#_Toc531955190)

[ГЛАВА 3. ДИАГРАММЫ КЛАССОВ 10](#_Toc531955191)

[ЗАКЛЮЧЕНИЕ 12](#_Toc531955192)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc531955193)

## **ВВЕДЕНИЕ**

Разрабатываемое мобильное приложение предназначено для проведения качественных занятий по бегу. Данным ПО может пользоваться любой человек. Мобильное приложение позволяет спланировать режим тренировок, улучшить спортивные качества как новичков, так и более опытных людей.

Началом работы с данной системы является скачивание и регистрация, а после следует создать свой режим, либо использовать уже созданные.

Подсистема предназначена для решения следующих задач:

* Улучшение физических качеств человека.
* Проведение тренировок по лёгкой атлетике согласно спланированному режиму.
* Обучение правильному выполнению упражнений и технике бега.

Объектом исследования является процесс разработки мобильного приложения для планирования индивидуальных тренировок по лёгкой атлетике.

Исследования выполняются путём построения следующих моделей:

* Функциональной (IDEF0).
* Потоков данных (DFD).
* Диаграмма классов (UML).

Целью моделирования является наглядное представления всех процессов разработки ПО начиная с задумки, заканчивая готовым продуктом. Данное представление поможет разобраться со всем запланированным функционалом разрабатываемого ПО, учесть все нюансы разработки, которые могут возникнуть.

Функциональная модель разрабатывается с точки зрения владельца продукта, который непосредственно занимается созданием запланированного мобильного приложения для планирования индивидуальных тренировок по лёгкой атлетике.

## **ГЛАВА 1. ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ (IDEF0)**

Функциональная модель – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов [1]. Особенностью данной моделью является упор на самоподчинённость объектов, также позволяет описать все процессы с достаточной точностью.

В IDEF0 все данные делятся на 4 различных типа, а именно:

* Внешние входные информационные потоки.
* Внешние выходные информационные потоки.
* Внешние управляющие потоки.
* Механизмы.

Внешними входными информационными потоками процесса разработки мобильного приложения для планирования индивидуальных тренировок по лёгкой атлетике являются:

* Информация о предметной области.

Выходными информационными потоками процесса являются:

* Готовое автоматизированное мобильное приложение.

Внешними управляющими потоками процесса являются:

* Заказчик.
* ГОСТ.

Основными механизмами процесса являются:

* Аналитик.
* Разработчик.
* IDE.

На рисунках 1-5 представлены диаграммы IDEF0, где 3 блока A1, A2, A3 декомпозируются.

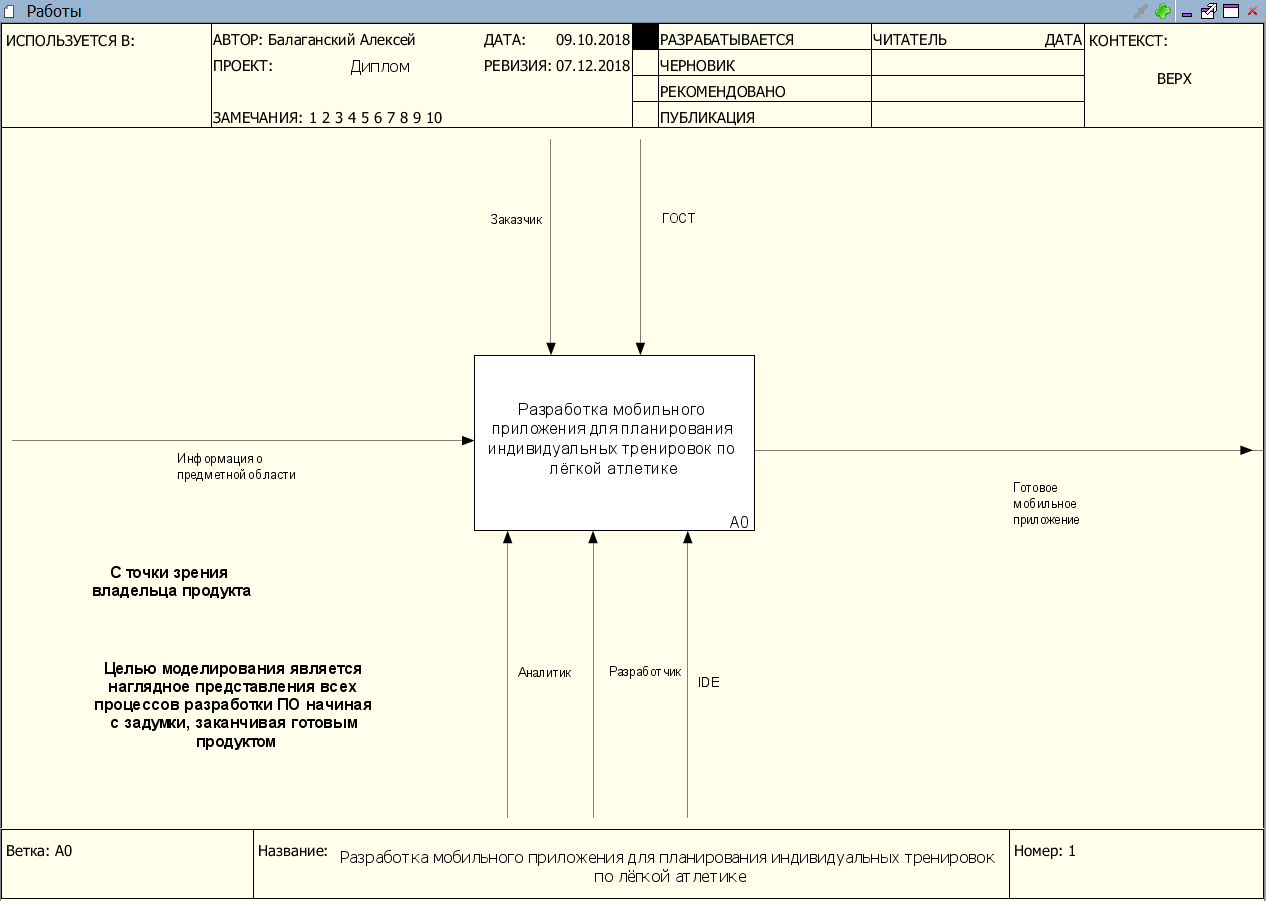


Рис. 1. Контекстная диаграмма

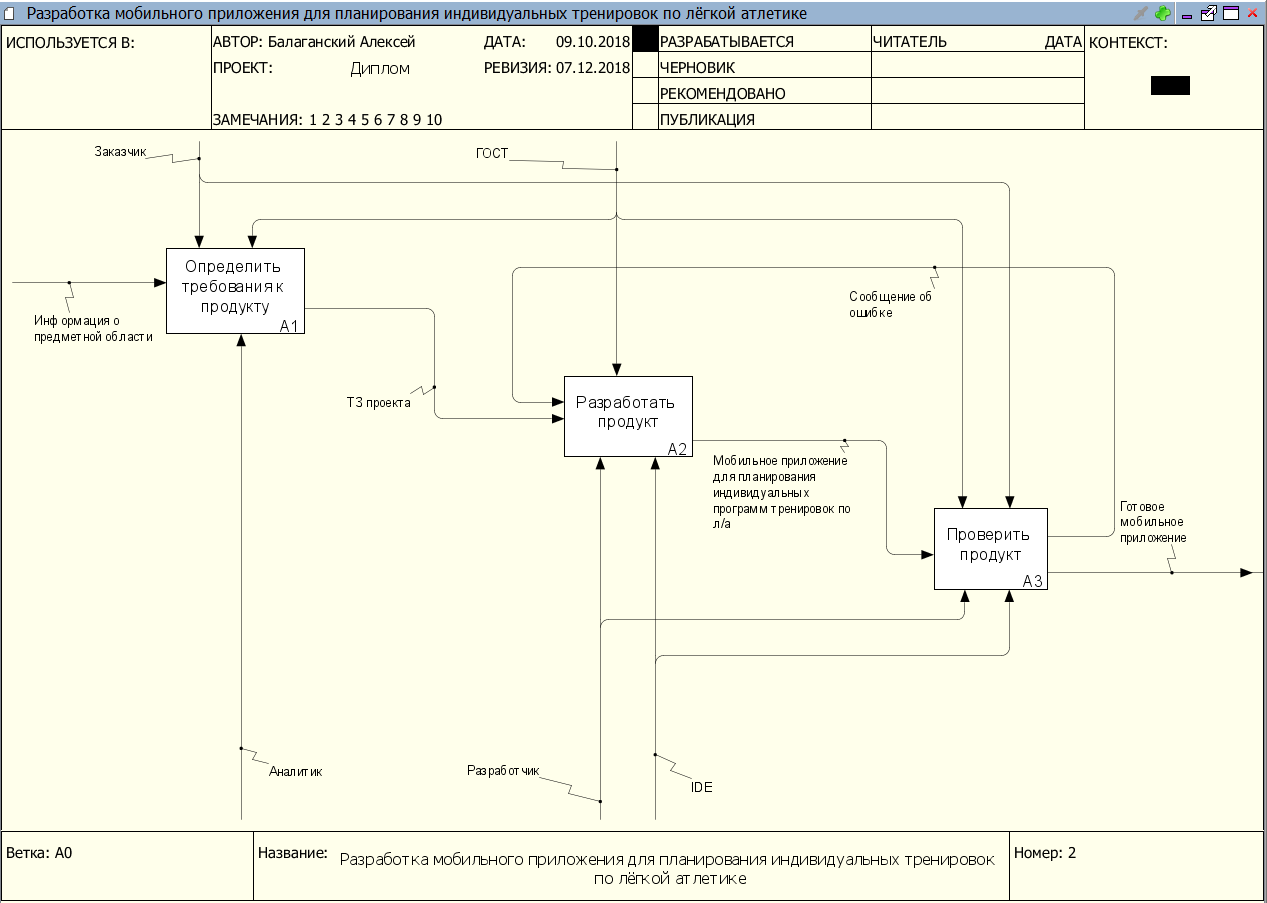


Рис. 2. Диаграмма процессов

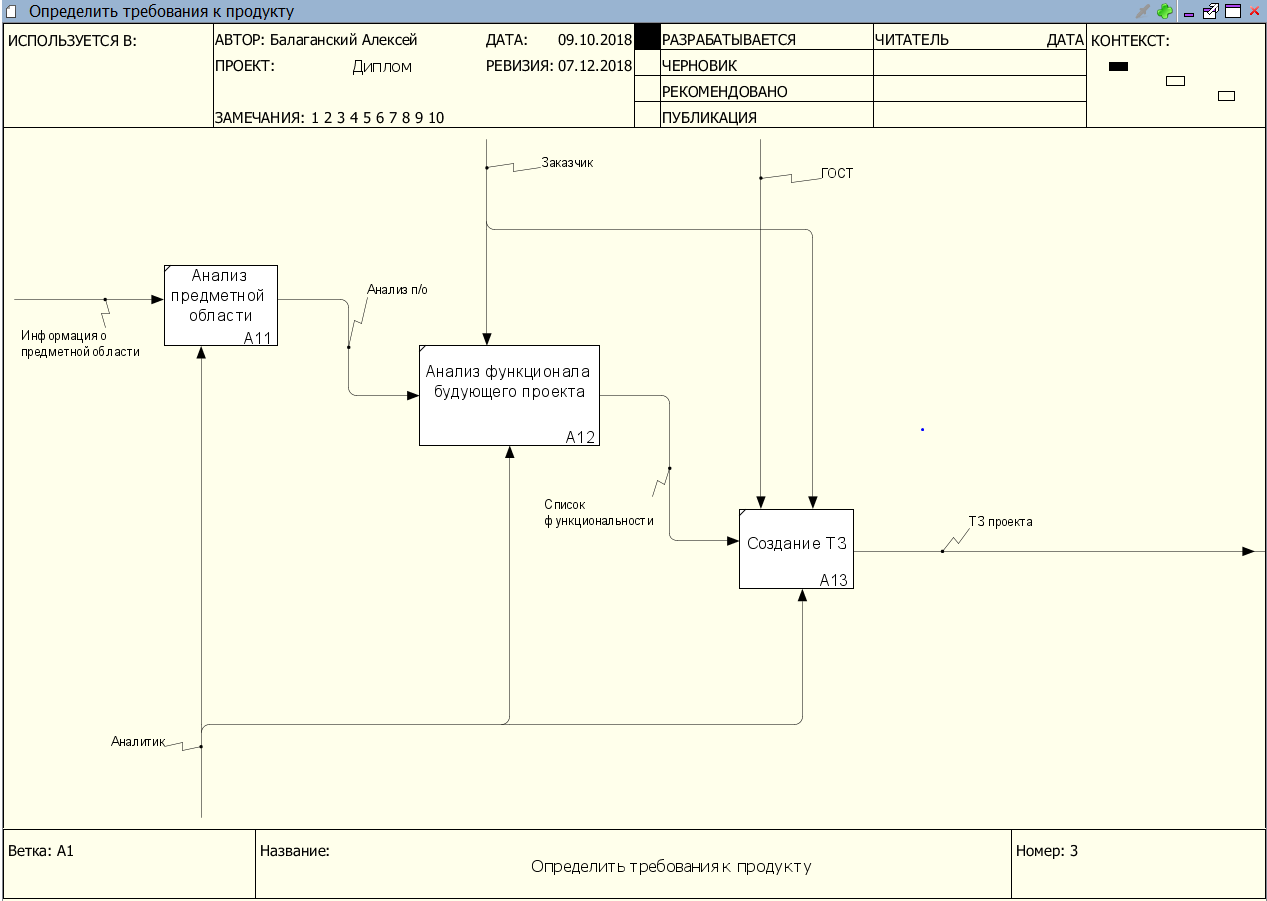


Рис. 3. Диаграмма процессов управления

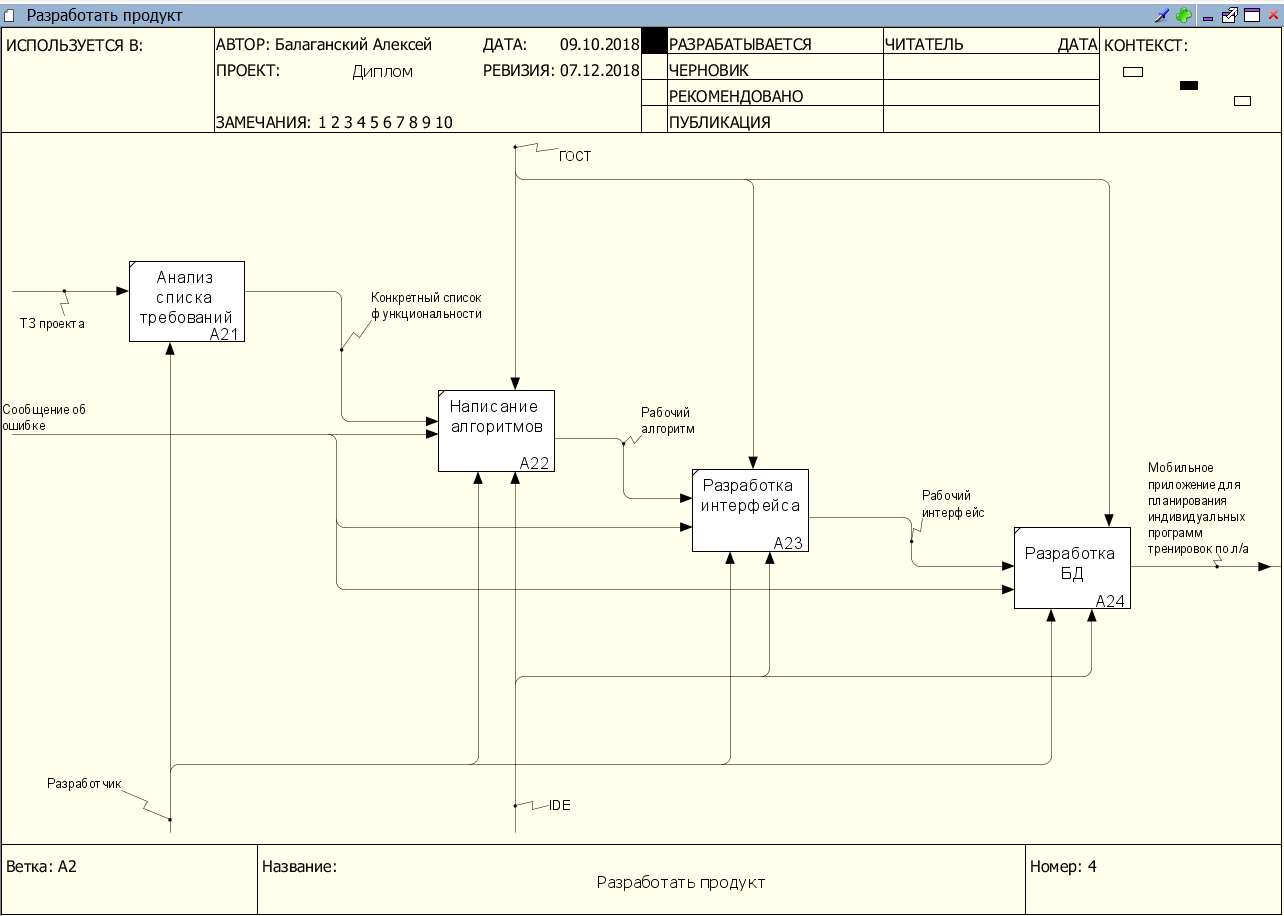


Рис. 4. Диаграмма процессов разработки

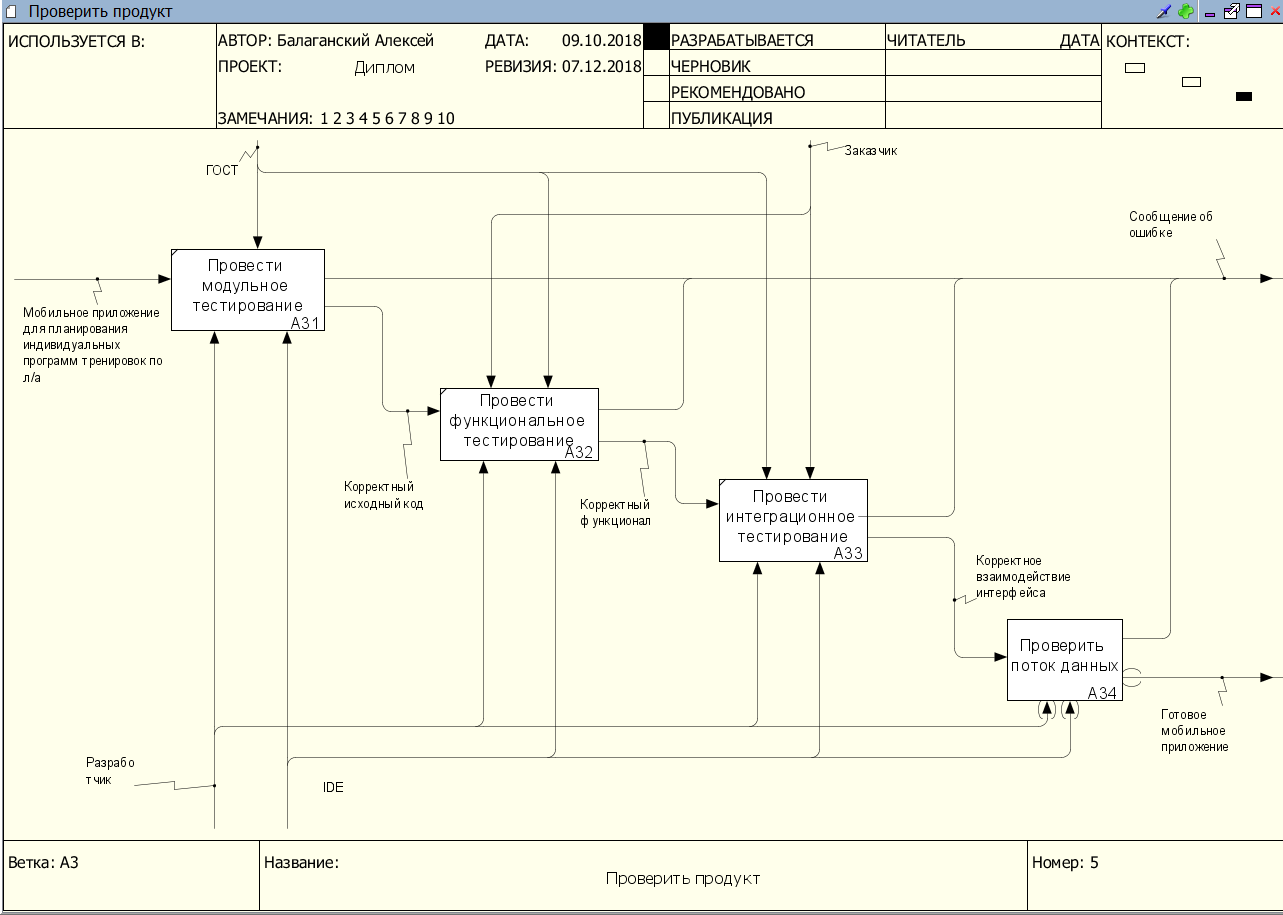


Рис. 5. Диаграмма процессов проверки

## **ГЛАВА 2. МОДЕЛЬ ПОТОКОВ ДАННЫХ (DFD)**

Целью диаграммы DFD является демонстрация, как каждый процесс преобразует свои входные данные в выходные, а также позволяет выявить отношения между процессами [2].

Наименования объектов собственной базы данных информационной

системы приводятся в формате «БД.Таблица».

В процессе декомпозиции функциональных блоков была выделена 1 диаграмма потоков данных (рис. 6). На данной диаграмме 1-3 блоки являются экранными формами, а 4 модулем обработки.

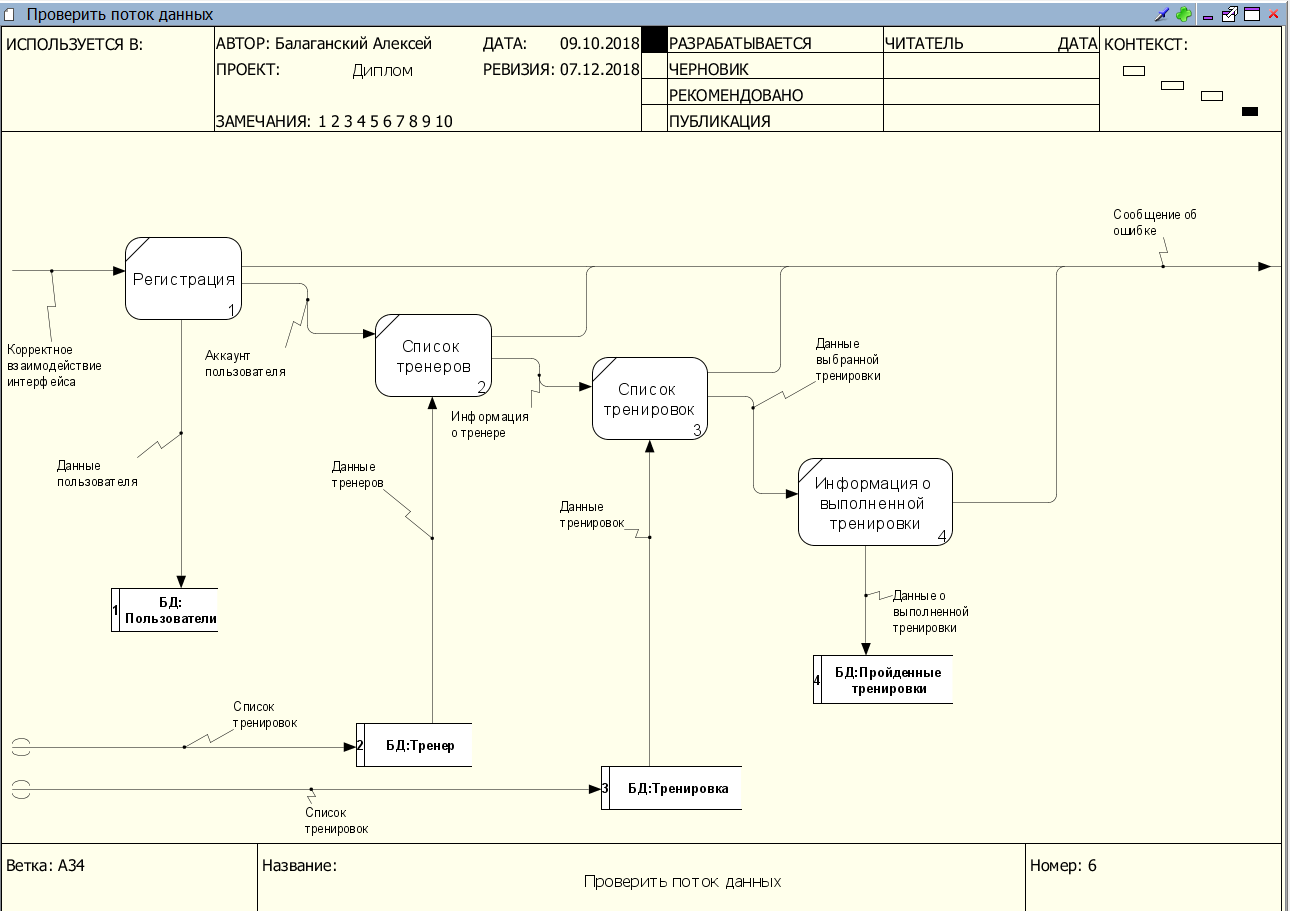


Рис. 6. Диаграмма потоков данных проверки работоспособности БД

Расчёт не выровненных функциональных точек приведены в таблице 1.

Таблица 1.

**Расчёт UFP**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Номер** | **Наименование** | **Определение** | **Форм** | **Данных** | **UFP** |
|  | Разработка мобильного приложения для планирования индивидуальных тренировок по лёгкой атлетике | Система позволяет составить индивидуальный план тренировок по лёгкой атлетике |  |  |  |
| A1 | Определить требования к продукту | Этап анализа предметной области, и постановка ТЗ | 0 | 0 | 0 |
| A2 | Разработать продукт | Этап разработки продукта | 0 | 0 | 0 |
| A3 | Проверить продукт | Этап проверки работы подсистемы, а именно работоспособность без ошибок и совпадение функционала в соответствие с ТЗ | 4 | 4 | 44 |
| Всего |  |  |  |  | 44 |

## 

## **ГЛАВА 3. ДИАГРАММЫ КЛАССОВ**

Диаграмма классов (англ. Static Structure diagram) - структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов(полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования [3].

В курсовой работе были рассмотрены 3 диаграммы классов: для потоков (рис. 7), для модулей (рис. 8) и для ролей (рис. 9).

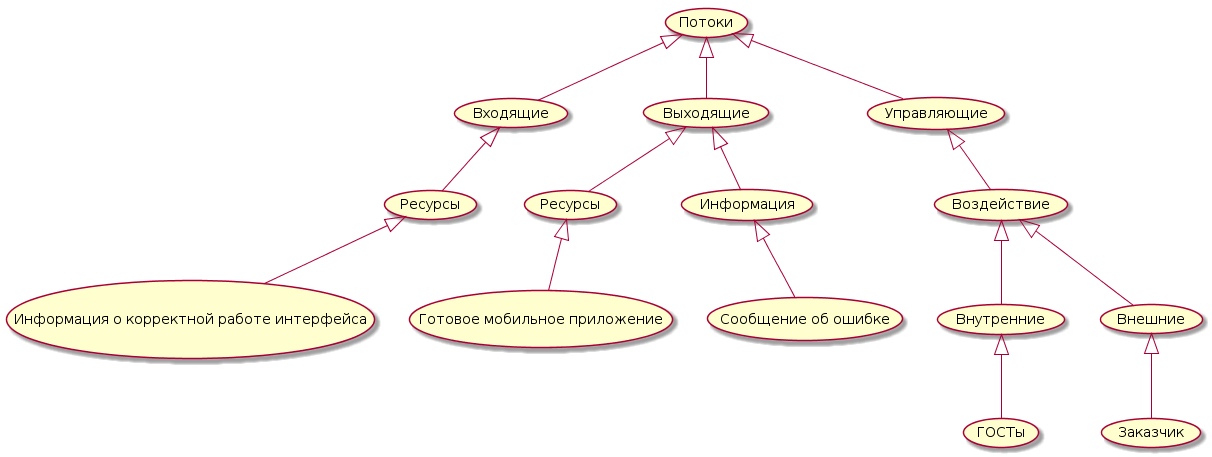


Рис. 7. Диаграмма классов для потоков



Рис. 8. Диаграмма классов для модулей

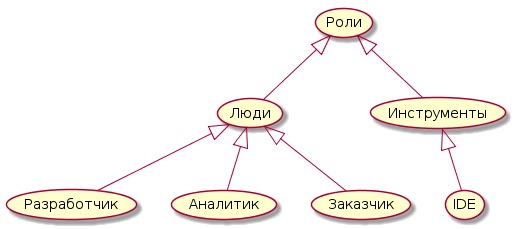


Рис. 9. Диаграмма классов для ролей

## **ЗАКЛЮЧЕНИЕ**

Таким образом, в ходе курсового проекта были созданы модели для разработки мобильного приложения по планированию индивидуальных тренировок о лёгкой атлетике. Была составлена диаграмма IDEF0, которая имела 3 уровня декомпозиции, и 1 диаграмму потоков данных DFD. Эти диаграммы помогают.

Также в результате курсовой работы был высчитан эффект от проекта, который будет заключатся в увеличении времени наработки на отказ, что даёт большую пропускную способность и позволяет обрабатывать больше получаемых запросов. Сравнение считалось по количество тренировок, которая могла создать система за месяц и общее время работы системы учитывая показатели, изображённые на рис. 10, а все расчёты приведены в таблице 2.

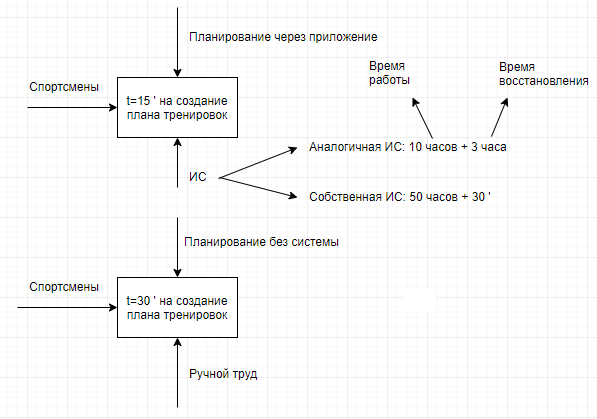


Рис. 10. Расчёт эффекта от проекта

Таблица 2

Расчёт эффекта от проекта

|  |
| --- |
| Период рассмотрения = 1 месяц = 30.4 \* 24 = 729,6 часов=43766 минут.  Рассмотрение полностью работоспособной системы: 43766 минут / 15 мин (время на создание плана тренировок с помощью системы) = 2918 тренировок, которая может создать система, без прерываний.  Рассмотрим количество тренировок для аналогичной ИС:  10 часов= 600 минут.  600 минут / 15 мин + 6 (количество тренировок во временя простая, используя ручной труд) = 46 тренировок за 13 часов.  729,6 / 13 \* 46 = 2581 тренировка за месяц, которые могут быть созданы, что составляет 88% (2581/2918).  729,6 - 729,6 / 13 \* 3 = 561 часов работоспособности системы, что составляет 76%.  Рассмотрим количество тренировок для созданной ИС:  50 часов= 3000 минут.  3000 минут / 15 мин + 1 (количество тренировок во временя простая, используя ручной труд) = 201 тренировок за 50.5 часов.  729,6 / 50.5 \* 201 = 2903 тренировка за месяц, которые могут быть созданы, что составляет 99% (2903/2918).  729,6 - 729,6 / 50.5 \* 0,5 = 722,1 часов работоспособности системы, что составляет 98%. |

Исходя из полученный результатов, можно сделать вывод, что создаваемая система создаёт за месяц на 11% больше тренировок, а также время работы на 22% выше аналога.

Расчеты, выполненные методом FPA IFPUG (рис. 10) на основании данных функциональной модели, позволяют оценить сложность требуемых для создания информационной системы программных средств в 41 выровненных функциональных точек (DFP), а объем программного кода на языках программирования высокого уровня - в 2046 строк кода.

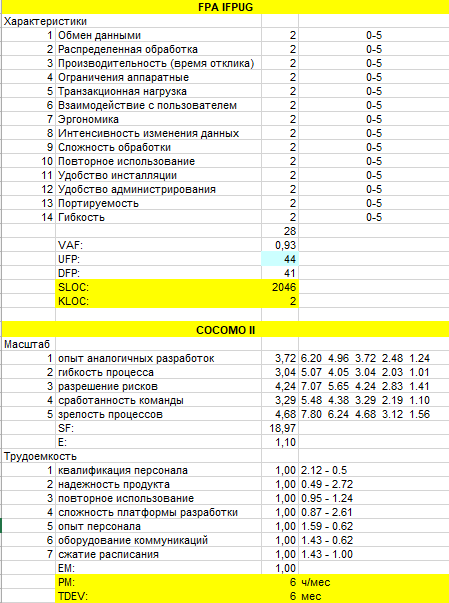


Рис. 10. Сроки создания разрабатываемой информационной системы

Расчеты, выполненные методом COCOMO II (рис. 11), позволяют оценить общие трудозатраты проекта разработки программных средств в 6 человеко-месяца, а ожидаемую продолжительность проекта – в 6 месяцев.

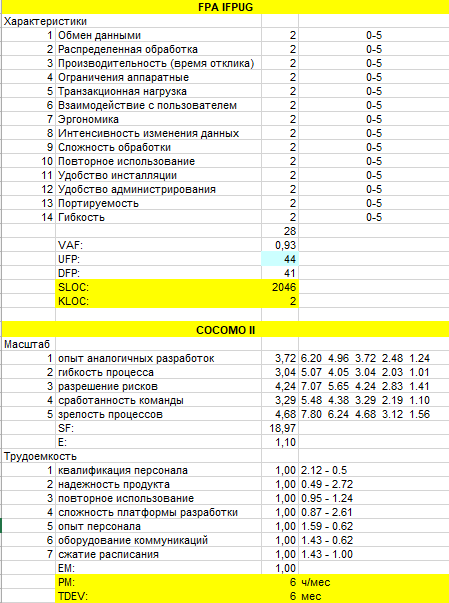


Рис. 11. Расчёт трудозатрат

Разрабатываемая система может применяться любым человеком, который хочет заняться лёгкой атлетикой. Данная система помогает спланировать оптимальную программу тренировок для наилучшего результата.

Сформированные модели будут использованы в выпускной квалификационной работе «Разработка мобильного приложения для автоматизации индивидуального планирования тренировок по лёгкой атлетике».

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1) IDEF0 [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/IDEF0.

2) DFD [Электронный ресурс]. URL: https://e-educ.ru/bd14.html.

3) UML [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki Диаграмма\_классов