**2 Проектирование программного обеспечения при структурном подходе**

**2.1 Проектирование функциональной модели IDEF0**

IDEF0, сокращение от "Integration Definition for Function Modeling", представляет собой стандартизированную методологию функционального моделирования, которая нашла широкое применение в области управления бизнес-процессами. Эта методология обеспечивает структурированный подход к анализу и оптимизации различных систем и процессов.

Основной принцип IDEF0 заключается в том, что система или процесс разбивается на функциональные блоки, каждый из которых выполняет конкретную функцию. Эти функциональные блоки представляются в виде диаграмм, которые иллюстрируют входы и выходы каждого блока, а также связи между ними. Такой подход позволяет легче понимать внутреннюю структуру системы и поток информации между её элементами.

Иерархическая природа IDEF0 позволяет строить модели различных уровней детализации. На более высоком уровне абстракции отображаются общие функции системы, в то время как на более низком уровне представлены более детальные аспекты функций. Такой подход позволяет анализировать систему как целое и одновременно углубляться в детали её работы.

IDEF0 не только помогает визуализировать функции и процессы системы, но и способствует выявлению узких мест, оптимизации бизнес-процессов и принятию обоснованных управленческих решений. Графическая нотация делает сложные концепции более доступными и понятными для различных участников процесса анализа и управления. [2]

Контекстная диаграмма IDEF0 представлена на рисунке 2.1.

Входными данными на контекстной диаграмме являются «Задание», «Отметка задания», «Запрос на сохранение».

Выходными данными являются «Сохраненное задание» и «Статус задания».

Механизмами являются «Хранилище данных», «Система и пользователь».

Управлением является «Требования системы».

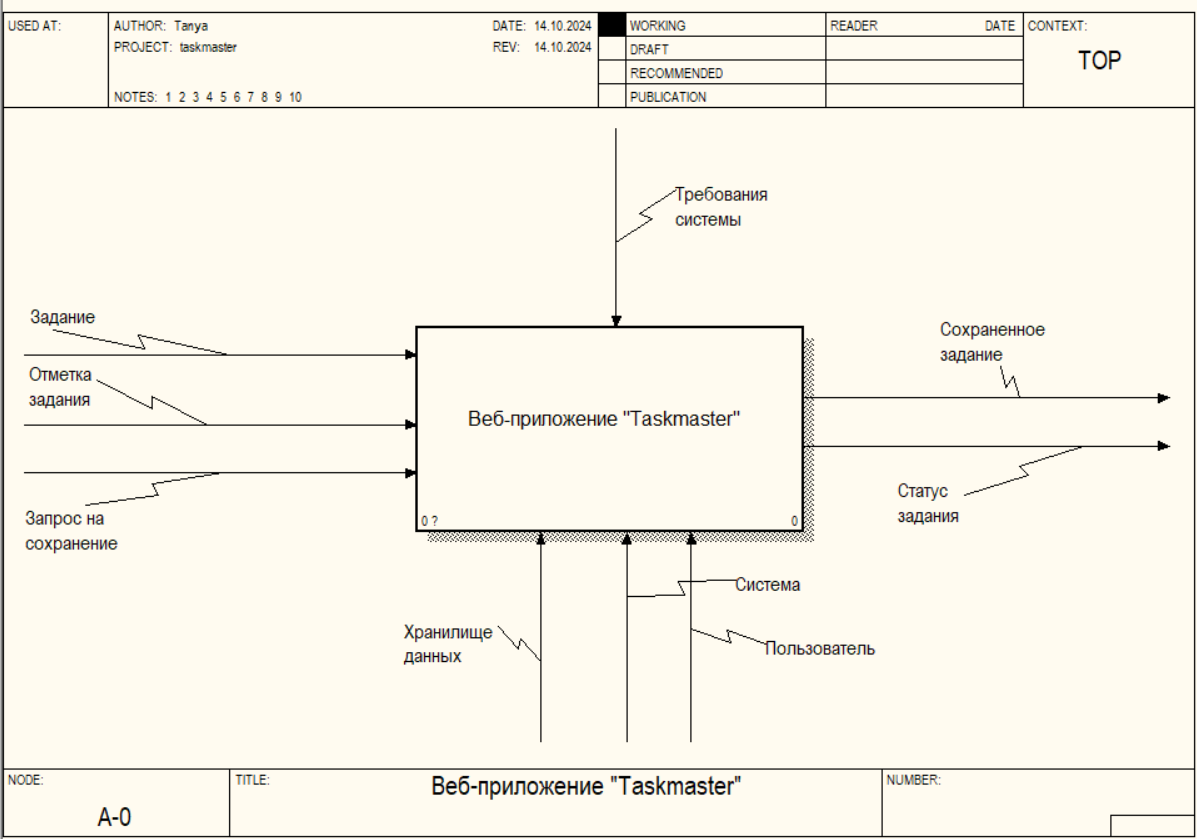


Рисунок 2.1 – Контекстная диаграмма IDEF0

Источник: собственная разработка

Для данной контекстной диаграмма была выполнена декомпозиция первого уровня. Декомпозиция первого уровня представлена на рисунке 2.2.

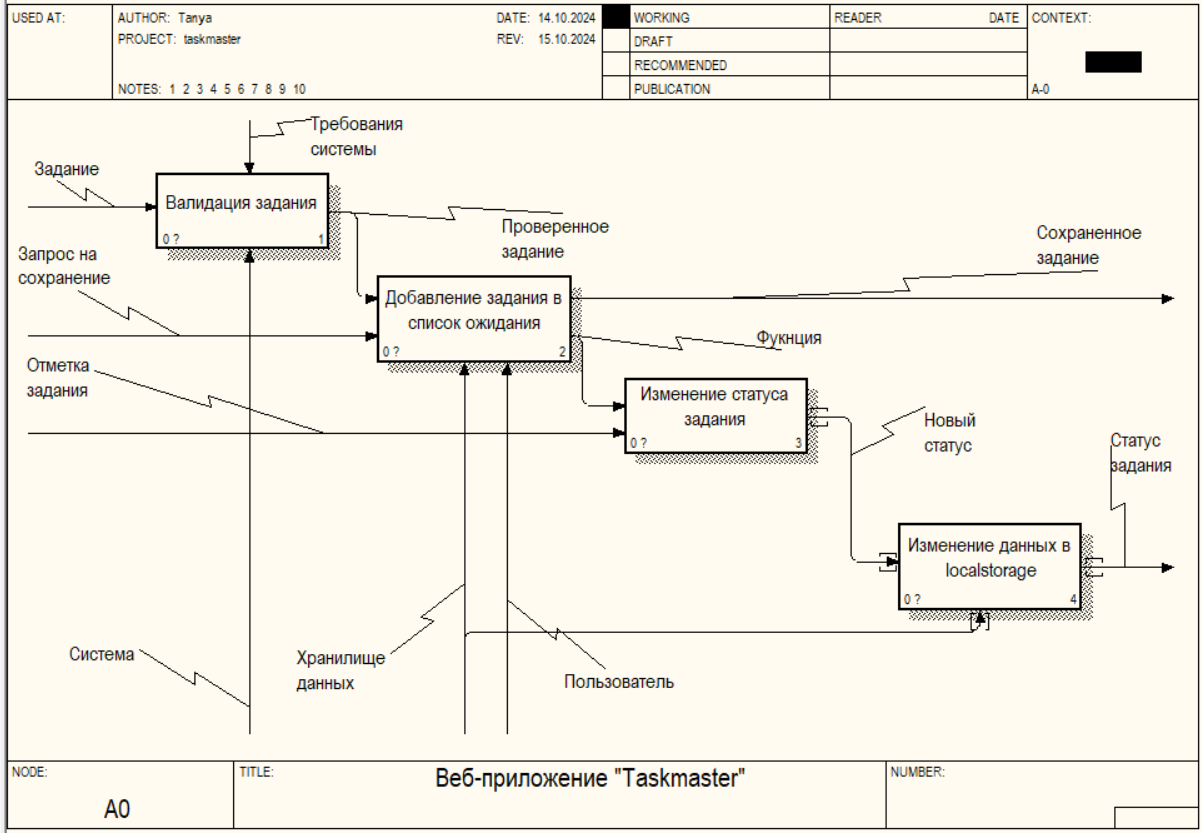


Рисунок 2.2 – Декомпозиция первого уровня

Источник: собственная разработка

Функциональными блоками при декомпозиции первого уровня являются: «Валидация данных», «Добавления задания в список ожидания», «Изменение статуса задания» и «Изменение данных в LocalStorage».

Была выполнена декомпозиция второго уровня для функционального блока «Изменение статуса задания». Декомпозиция второго уровня представлена на рисунке 2.3.

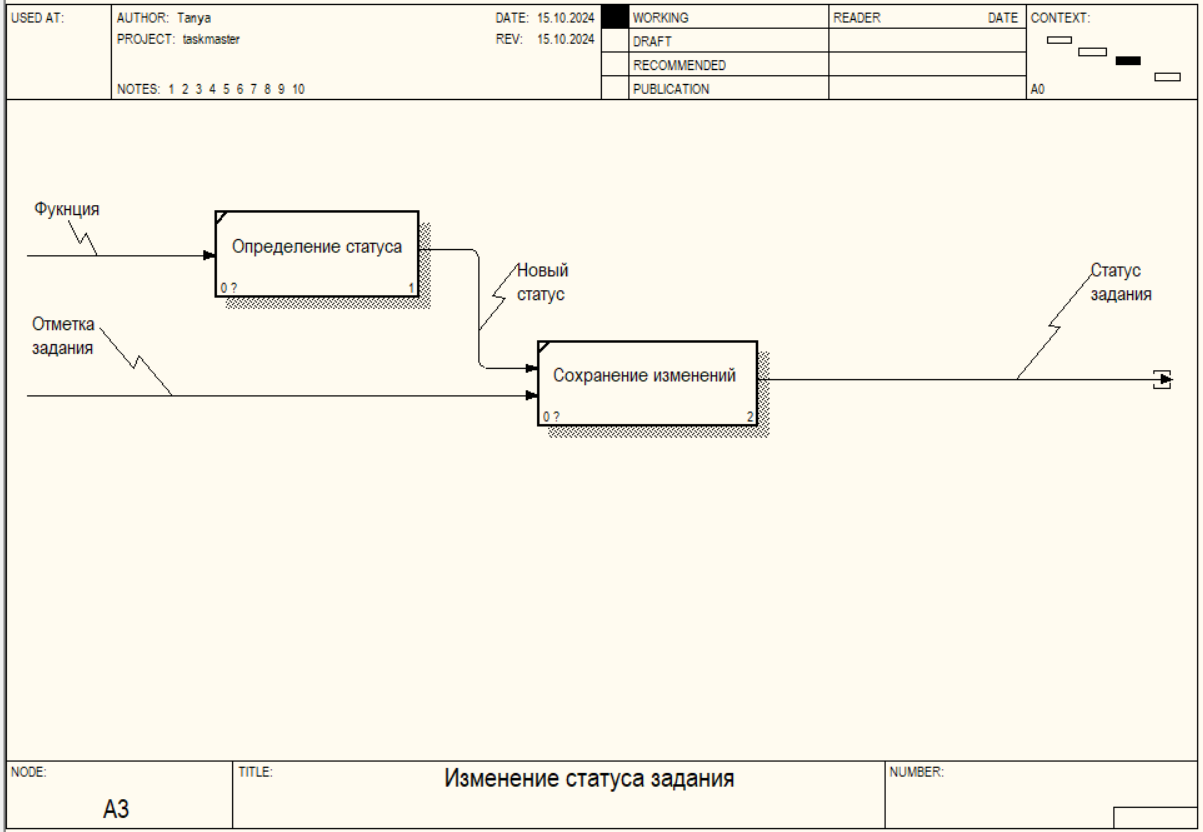


Рисунок 2.3 – Декомпозиция второго уровня

Источник: собственная разработка

На данной диаграмме функциональными блоками являются «Определение статуса» и «Сохранение изменений».

**2.2 Проектирование бизнес-процессов в нотации IDEF3**

IDEF3 является стандартом документирования технологических процессов, происходящих на предприятии, и предоставляет инструментарий для наглядного исследования и моделирования сценариев**.**

В отличие от IDEF0, представляющего моделируемую систему как совокупность видов деятельности, IDEF3 представляет собой технику моделирования деятельности как последовательности событий, а также участвующих в этих событиях объектов. Модели IDEF3 могут использоваться для детализации функциональных блоков IDEF0, которые более низкого уровня. IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые в дальнейшем могут быть использованы для имитационного анализа.

IDEF3 - это метод, имеющий основной целью дать возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе. Техника описания набора данных IDEF3 является частью структурного анализа. Каждая работа в IDEF3 описывает какой-либо сценарий бизнес-процесса и может являться составляющей другой работы.

Диаграмма IDEF3 может включать следующие элементы: работы, процессы, связи, функции и операции. [9]

Диаграмма IDEF3 для блока «Валидация задания» представлена на рисунке 2.4.

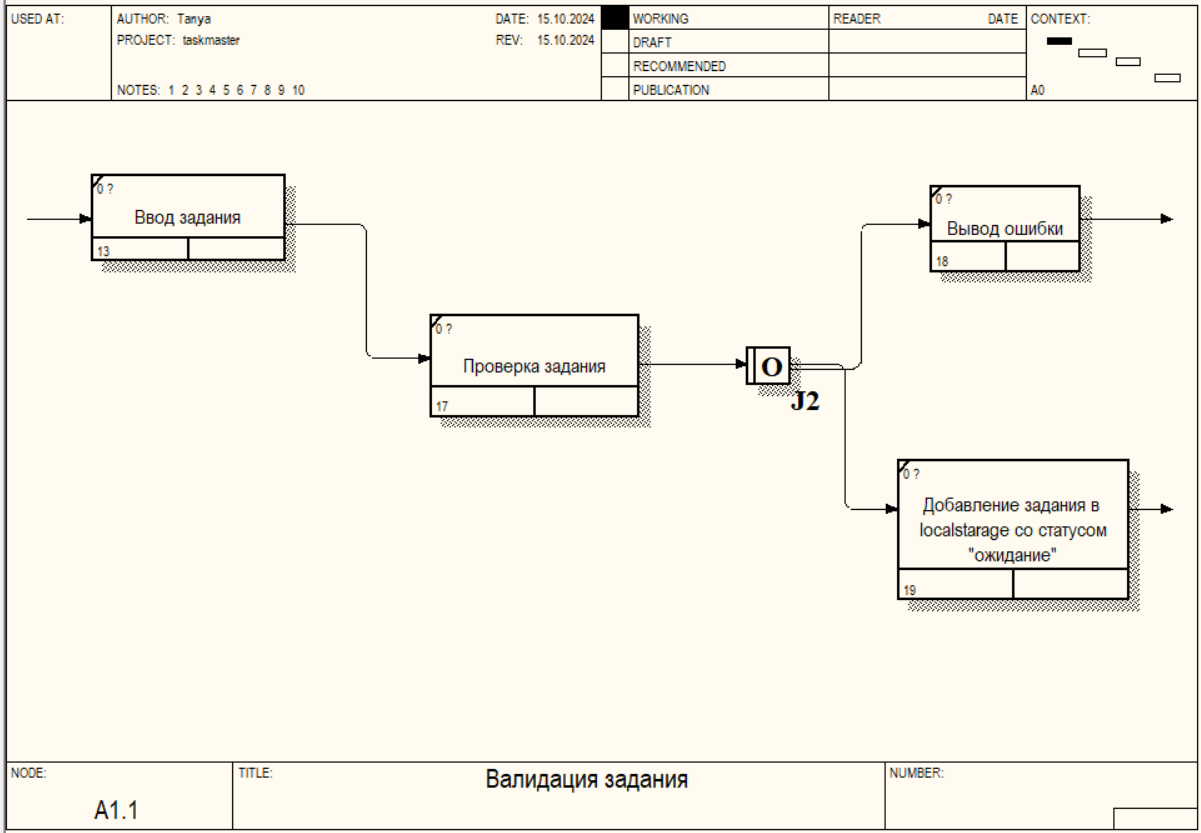


Рисунок 2.4 – Диаграмма IDEF3 для блока «Валидация задания»

Источник: собственная разработка

На данной диаграмме работами являются: «Ввод данных», «Проверка задания», «Вывод ошибок» и «Добавление задания в localstorage со статусом «ожидание»». Так же присутствует узел «OR», который является ветвлением.

Диаграмма IDEF3 для блока «Добавление задания в список ожидания» представлена на рисунке 2.5.

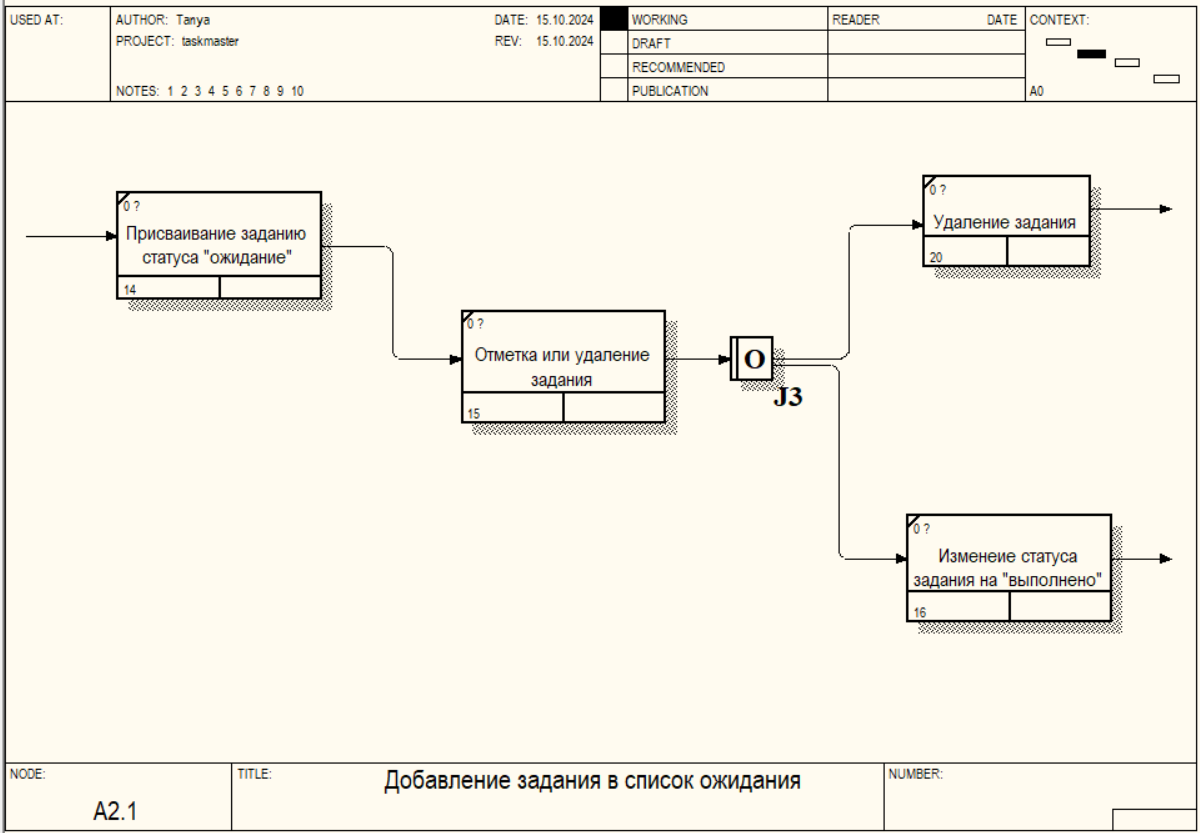


Рисунок 2.5 – Диаграмма IDEF3 для блока «Добавление задания в список ожидания»

На данной диаграмме работами являются «Присваивание заданию статуса «ожидание»», «Отметка или удаление задания», «Удаление задания» и «Изменение статуса задания на «выполнено»». Так же присутствует узел «OR», который является ветвлением.

**2.3 Проектирование функциональной модели DFD**

Диаграммы DFD, известные также как диаграммы потока данных, представляют собой визуальный инструмент для моделирования и анализа потоков данных в информационной системе. Данные диаграммы являются эффективным способом описания, как данные перемещаются через систему, указывая на входящие и исходящие потоки данных, а также места хранения.

На диаграмме DFD можно анализировать, как информация проходит через различные компоненты системы, отражая пути передачи данных между различными узлами, а также взаимосвязи между ними. Важно отметить, что простота и понятность являются ключевыми преимуществами диаграмм потоков данных, поскольку есть возможность четко визуализировать внутреннюю структуру системы.

Стандартные элементы, используемые на диаграммах DFD, включают прямоугольники для процессов или функций, круги для внешних сущностей, хранилищ данных или источников данных, а также стрелки для обозначения потока данных между этими элементами. Краткие текстовые метки часто используются для уточнения содержания каждого элемента на диаграмме.

DFD можно разделить на две основные категории. Логические диаграммы DFD являются абстрактными моделями, которые фокусируются на потоках данных, не углубляясь в конкретные реализации системы. В свою очередь, физические диаграммы DFD отображают конкретные устройства и ресурсы системы, описывая текущее состояние либо планируемую структуру.

Диаграмма DFD состоит из:

процесс — активность, которая приводит к преобразованию данных;

внешние сущности — это участники процесса, которые взаимодействуют с данными;

хранилище данных — место, где хранится информация для последующего использования системой;

потоки данных — маршруты, по которым информация перемещается между внешними сущностями, процессами и хранилищами данных. [3]

Контекстная диаграмма DFD представлена на рисунке 2.6.

Внешними сущностями на данной диаграмме являются «Пользователь» и «Система». Потоками данных являются «Отметка задания», «Задание», «Статус задания», «Запрос на сохранение» и «Проверенное задание».

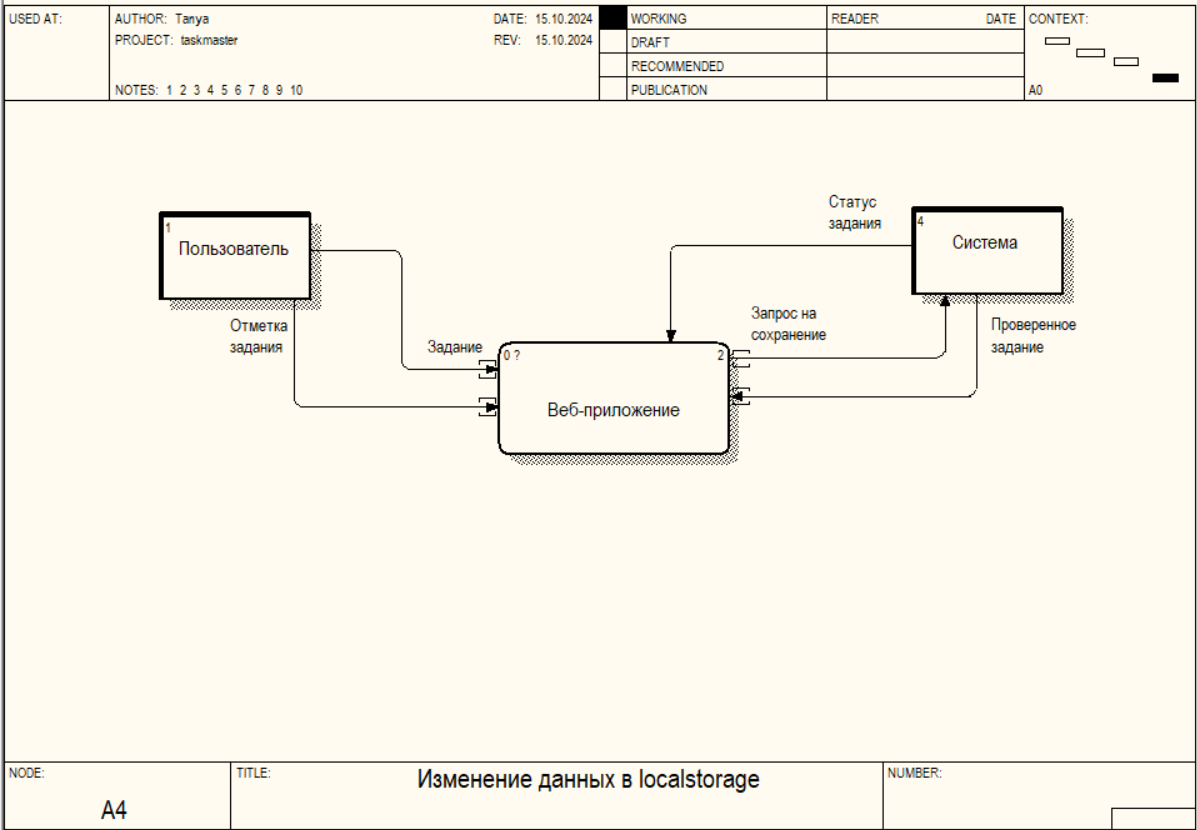


Рисунок 2.6 – Контекстная диаграмма DFD

Источник: собственная разработка

Для контекстной диаграммы была выполнена декомпозиция. Декомпозиция контекстной диаграммы DFD представлена на рисунке 2.7.

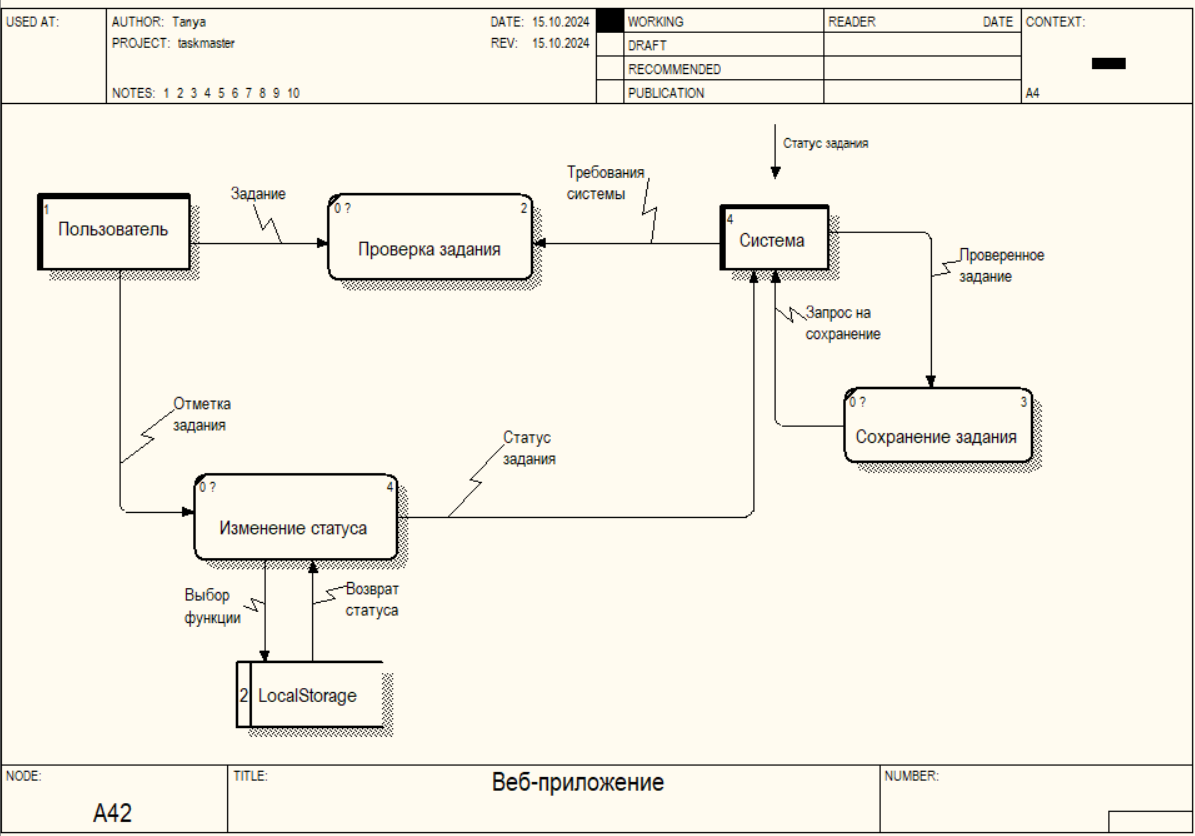


Рисунок 2.7 – Декомпозиция контекстной диаграммы DFD

Источник: собственная разработка

На данной диаграмме сущностями являются «Пользователь» и «Система». Функциональными блоками являются «Проверка задания», «Сохранение задания» и «Изменение статуса». Хранилищем данных является «LocalStorage».

**2.4 Проектирование логической модели данных IDEF1X**

Методология IDEF1X представляет собой эффективный метод разработки реляционных баз данных, который специально разработан для удобного построения концептуальной схемы. Основанный на концепции "сущность – связь", IDEF1X обеспечивает структурированное и наглядное представление данных, что делает процесс моделирования более понятным и эффективным.

Логическая модель в методологии IDEF1X представляет собой графическое отображение структуры базы данных, учитывая выбранную модель данных (например, иерархическую, сетевую, реляционную и т.д.). Модель описывает, какие данные хранятся в базе данных, включая объекты предметной области, атрибуты и связи между ними, но не затрагивает аспект "как" данные будут храниться.

В логической модели IDEF1X ключевыми элементами являются сущности и связи. Сущности делятся на два типа: независимые и зависимые. Независимая сущность обладает уникальным идентификатором, который не наследуется от других сущностей. С другой стороны, зависимая сущность имеет уникальный идентификатор, включающий по крайней мере одну связь с другой сущностью.

Методология IDEF1X сфокусирована на проектировании реляционных моделей баз данных. Реляционная модель базируется на понятии нормализованных отношений, которые представлены в виде таблиц. Этот подход обеспечивает структурированное хранение и управление данными, что способствует эффективной работе с информацией в рамках базы данных, а также облегчает разработку и поддержку баз данных в долгосрочной перспективе. [4]

Модель базы данных, представленная с помощью методологии IDEF1X, представлена на рисунке 2.7.

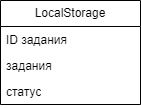


Рисунок 2.7 – Модель базы данных

Источник: собственная разработка

Данная модель имеет одну сущность «LocalStorage», которая является независимой. Данная сущность имеет атрибуты: «ID задания», «задания», «статус».

Основные сущности представлены в таблице 2.1.

Таблица 2.1 – Основные сущности

|  |  |  |
| --- | --- | --- |
| № | Наименование сущности | Краткое описание |
| 1 | LocalStorage | Хранилище данных. |

Источник: собственная разработка

Описание атрибутов сущности «LocalStorage» представлены в Таблице 2.

Таблица 2.2 – Описание атрибутов сущности «LocalStorage»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ключевое поле | Название | Назначение |
| 1 | ПК(Первичный ключ) | ID задания | Ключевое поле предназначено для однозначной идентификации записи в таблице. |
| 2 |  | Задания | Текст задания. |
| 3 |  | Статус | Статус задания. |

Источник: собственная разработка

IDEF0, DFD и IDEF1X - это различные типы диаграмм, которые используются в системном анализе и проектировании для моделирования бизнес-процессов, структуры данных и информационных потоков в системе.