

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**СОГЛАСОВАНО**

Старший преподаватель департамента  
программной инженерии факультета  
компьютерных наук

\_\_\_\_\_ С. А. Шершаков  
«\_\_» \_\_\_\_\_ 2020 г.

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия»  
профессор департамента программной  
инженерии, канд. техн. наук

\_\_\_\_\_ В.В. Шилов  
«\_\_» \_\_\_\_\_ 2020 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.04.01-01 81 01-1

**Плагин для платформы IntelliJ для мониторинга процесса создания  
программы и формирования отчета**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.01-01 81 01-1-ЛУ**

Исполнитель  
студент группы \_\_\_\_\_

\_\_\_\_\_ / Т. В. Тибилов /  
«\_\_» \_\_\_\_\_ 2020 г.

**Москва 2020**

УТВЕРЖДЕН  
RU.17701729.04.01-01 81 01-1-ЛУ

**Плагин для платформы IntelliJ для мониторинга процесса создания  
программы и формирования отчета**

**Пояснительная записка**

**RU.17701729.04.01-01 81 01-1**

**Листов 20**

<i>Инв. № подл</i>	<i>Подп. и дата</i>	<i>Взам. инв. №</i>	<i>Инв. № дубл.</i>	<i>Подп. и дата</i>
RU.17701729.04.01-01 81 01-1				

**Москва 2020**

## АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к программе «task\_manager-1.0» («Плагин для платформы IntelliJ для мониторинга процесса создания программы и формирования отчета»), предназначенной для создания и управления задачами проекта и генерации отчета о проделанной работе.

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

- постановка задачи на разработку программы;
- описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
- описание и обоснование выбора метода организации входных и выходных данных;
- описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [1];
- 2) ГОСТ 19.102-77 Стадии разработки [2];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [3];
- 4) ГОСТ 19.104-78 Основные надписи [4];
- 5) ГОСТ 19.105-78 Общие требования к программным документам [5];
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
- 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей пояснительной записки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## СОДЕРЖАНИЕ

<b>1. ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>1.1. Наименование программы .....</b>	<b>4</b>
<b>1.2. Документы, на основании которых ведется разработка .....</b>	<b>4</b>
<b>2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ .....</b>	<b>5</b>
<b>2.1. Назначение программы .....</b>	<b>5</b>
<b>2.1.1. Функциональное назначение .....</b>	<b>5</b>
<b>2.1.2. Эксплуатационное назначение .....</b>	<b>5</b>
<b>2.1.3. Область применения программы .....</b>	<b>5</b>
<b>3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....</b>	<b>6</b>
<b>3.1. Постановка задачи на разработку программы .....</b>	<b>6</b>
<b>3.2. Описание алгоритма и функционирования программы .....</b>	<b>6</b>
<b>3.2.1. Описание алгоритма программы .....</b>	<b>6</b>
<b>3.2.2. Регистрация объектов-менеджеров и событий в IDE .....</b>	<b>6</b>
<b>3.2.3. Алгоритмы и классы базового модуля .....</b>	<b>6</b>
<b>3.2.4. Алгоритмы и классы модуля UI .....</b>	<b>8</b>
<b>3.2.5. Алгоритмы и классы модуля Action .....</b>	<b>10</b>
<b>3.3. Обоснование выбора алгоритма решения задачи .....</b>	<b>11</b>
<b>3.4. Возможные взаимодействия программы с другими программами .....</b>	<b>11</b>
<b>3.5. Описание и обоснование выбора метода организации входных и выходных данных .....</b>	<b>12</b>
<b>3.6. Описание и обоснование выбора состава технических и программных средств .....</b>	<b>12</b>
<b>4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ .....</b>	<b>13</b>
<b>4.1. Предполагаемая потребность .....</b>	<b>13</b>
<b>4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами .....</b>	<b>13</b>
<b>5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ .....</b>	<b>14</b>
<b>ПРИЛОЖЕНИЕ 1 .....</b>	<b>15</b>
<b>ПРИЛОЖЕНИЕ 2 .....</b>	<b>16</b>
<b>ПРИЛОЖЕНИЕ 3 .....</b>	<b>18</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**1. ВВЕДЕНИЕ****1.1. Наименование программы**

Наименование программы – «Плагин для платформы IntelliJ для мониторинга процесса создания программы и формирования отчета».

Наименование программы на английском языке – «Plugin for IntelliJ Platform for monitoring of developing process and report generation».

**1.2. Документы, на основании которых ведется разработка**

Разработка ведется на основании приказа Национального исследовательского университета «Высшая школа экономики» № 2.3-02/1112-04 от 11.12.2019 "Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук".

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 2.1. Назначение программы

#### 2.1.1. Функциональное назначение

Функциональным назначением программы является мониторинг процесса разработки программного продукта разработчиком/командой разработчиков и создание отчета на основе полученных данных. Плагин собирает информацию, которая предоставляется платформой для разработки программ IntelliJ, сохраняет соответствующие данные и формирует отчеты за установленный период времени в виде .CSV файлов. С помощью данной программы можно планировать дедлайны по задачам, планировать разработку проекта.

#### 2.1.2. Эксплуатационное назначение

Предполагается использование программного интерфейса при разработке ПО непосредственно заказчиками для анализа продуктивности выполнения заказа исполнителем. Программа будет полезна в случае создания фриланс-сервиса для мониторинга заказчиками процесса разработки продукта. Также возможно использование плагина в командных проектах менеджерами проектов. Основным преимуществом данной программы является её непосредственное внедрение в инфраструктуру IntelliJ, что позволяет быстро управлять задачами и создавать отчеты, не отвлекаясь от разработки быстро управлять задачами и создавать отчеты.

#### 2.1.3. Область применения программы

Программа носит прикладной характер. Программа подразумевает запись и сохранение отчетности о проделанной работе программистом (исполнителем) для последующего предоставления данных о процессе разработки продукта заказчику. Программа использует данные, предоставляемые платформой IntelliJ для мониторинга информации о процессе разработки и создания соответствующих записей о проделанной работе.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

#### 3.1. Постановка задачи на разработку программы

Постановка задачи – разработка плагина для открытой платформы разработки ПО IntelliJ, который позволит управлять задачами разработки проектов, автоматически отсчитывать и запоминать время, отведенное под реализацию задач, автоматически формировать отчеты по проделанной работе за указанный период. Плагин также должен представлять собой платформу для дальнейшего расширения функциональности последнего.

#### 3.2. Описание алгоритма и функционирования программы

##### 3.2.1. Описание алгоритма программы

Так как сложность программы по большей части составляют не алгоритмы, а непосредственно сама архитектура, то далее ей будет уделено большее внимание. Программа состоит из трех пакетов-компонент:

- базового модуля, содержащего классы и методы для работы бизнес-логики приложения;
- модуля, содержащего классы и методы, формирующие пользовательский интерфейс программы и взаимодействующие с бизнес-моделью;
- модуля, содержащего классы и методы, взаимодействующие с платформой IDE IntelliJ и обрабатывающие определенную информацию о текущем состоянии проекта.

##### 3.2.2. Регистрация объектов-менеджеров и событий в IDE

Для корректной работы плагина все необходимые менеджеры-синглтоны, а также события регистрируются в файле `plugin.xml`, в котором группируются соответственно архитектуре самого плагина. Так как большая часть архитектуры самой платформы IDE построена на регистрации и вызовах событий, а также на создании объектов-сервисов (объектов-менеджеров), то и данный плагин не стал исключением (далее будут рассмотрены все менеджеры данного плагина).

##### 3.2.3. Алгоритмы и классы базового модуля

Базовый модуль (*org.taimuraztibilov.taskmanager.base*) содержит в себе классы, представляющие основную бизнес-модель приложения. Классы *Project*, *Milestone*, *Task*, *KeyPoint*, *Label* представляют собой модели базы данных SQLite, ER-диаграмма которой представлена ниже (Рисунок 1).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

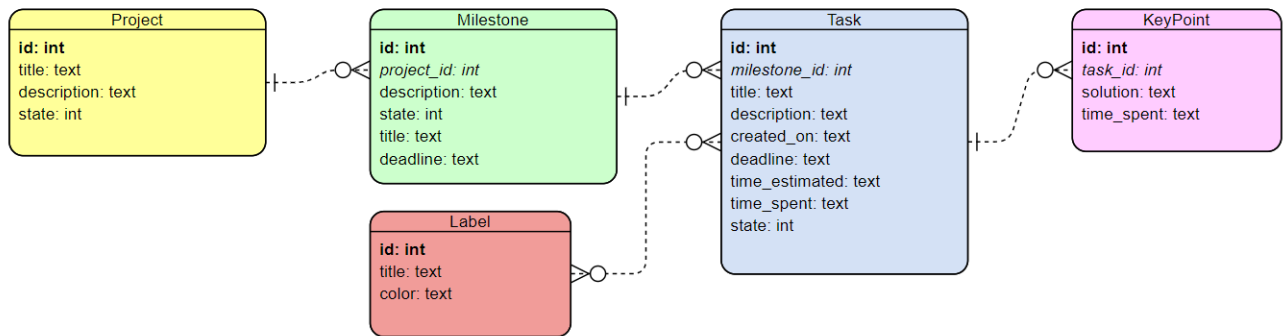


Рисунок 1. ER-диаграмма базы данных

Все сущности и поля базы данных реализованы по аналогии GitLab (в том числе и для последующего расширения в сторону последнего), который на уровне ведения репозитория предоставляет похожий функционал.

Для реализации функциональной части плагина используются четыре синглтона-менеджера:

1. *DataBaseManager*. Его основная задача – взаимодействие с базой данных плагина. Он также реализует интерфейс *DataEditor*, который позволяет классам-моделям оповещать об изменении значений в своих полях. С помощью JDBC SQLite3 создается подключение к базе данных плагина, в случае необходимости создаются недостающие таблицы. Также в данном классе существуют все необходимые методы для создания, изменения, получения и удаления соответствующих данных из БД.
2. *TimeManager*. Основная задача данного синглтона – это отслеживание времени выполнения выбранной задачи, а также для создания ключевых точек по задаче с описанием проделанной работы за один сеанс разработки в IDE. Далее данные, полученные с его помощью, будут доступны для формирования отчетов.
3. *ReportManager*. Этот синглтон выполняет функцию автоматического формирования отчета о проделанной работе за заданный период времени. Он обращается к базе данных с join-SQL запросом, в котором собирает все ключевые точки задач, закрытых в данном периоде. Сам запрос имеет следующий вид:

```

select t.title_m, t.title_t, k.date_closed, k.solution, k.time_spent
from keypoint k
inner join
(select ta.id, ta.title as title_t, m.title as title_m
  
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

from task ta
inner join milestone m on ta.milestone_id = m.id and m.project_id =
projectId ) t
on t.id = k.task_id
where date(k.date_closed) >= date( from ) and
date(k.date_closed) <= date(to)
order by 1, 2, 3

```

С помощью открытой библиотеки openCSV превращает полученные данные в .csv файл, который будет записан в корневой директории текущего проекта, открытого в IDE. Пример отчета можно увидеть на Рисунке 2.

Отчет по проекту с 2020-04-30 по 2020-05-30 ▼				
Наименование проекта:				
Разработчик:				
Организация:				
Веха	Задача	Дата	Комментарий	Затраченное время

Рисунок 2. Пример шаблона отчета

4. *PluginManagerService*. Данный менеджер оперирует основными данными, с которыми в данный момент пользователь взаимодействует через IDE. Он по сути дела является вспомогательным синглтоном, содержащем по большей части метаданные.

Все четыре вышеперечисленных класса регистрируются в платформе IntelliJ как сервисы платформы – это позволено сделать синглтонам, чтобы они подгружались класслоудером самой платформы при открытии программы IDE, и существовали на протяжении всего жизненного цикла программы вплоть до завершения работы в ней. В остальных же случаях классы и объекты будут подгружаться только после происхождения какого-либо из отмеченных событий и будут существовать только пока они определены платформой как используемые.

### 3.2.4. Алгоритмы и классы модуля UI

Модуль пользовательского интерфейса (*org.taimuraztibilov.taskmanager.ui*) содержит в себе статические методы, которые позволяют взаимодействовать пользователю с методами, доступными в базовом модуле. На примерах работы и взаимодействия с данными формами можно подробно рассмотреть функционал, доступный в методах базового класса.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Для начала стоит пояснить, что в случае создания окон с пользовательским интерфейсом не все так однозначно, как может показаться на первый взгляд. Дело в том, что базовый фреймворк для создания форм графического интерфейса *Swing* является очень гибким, но и в то же время легковесным, что накладывает некоторые ограничения на создания пользовательских элементов интерфейса.

Здесь для создания последних были использованы конструкторы компонентов, используемых в самой IDE, что являются *wtarper* классами над стандартными компонентами, что в свою очередь позволяет более детально построить интерфейс и довесить дополнительных элементов. Подробное описание компонентов JB, соответствующих компонентам из базового фреймворка, можно найти на странице документации по дизайн-коду IDE.

При работе с задачами их необходимо сначала создать. Для постройки пользовательских форм используется класс *AddDataFormBuilder*, который в свою очередь содержит соответствующие статические методы, формирующие окна для создания новых записей в БД плагина. Для этого полученные данные из полей интерпретируются в нужном формате, а затем передаются на вход *add*-методам экземпляра *DataBaseManager*.

Для выбора задачи, которую пользователь хочет отслеживать, в статических методах класса *ShowDataFormBuilder* строится 3 разных формы – для выбора проекта, вехи и непосредственно самой задачи. После каждого выбора в полях текущего выбора в *PluginManagerService* запоминаются *id* элементов, выбранных для отслеживания. Если до этого пользователь отслеживал другую задачу, программа предложит ему зафиксировать точку в разработке. При этом время, затраченное на задачу, *автоматически* подтянется к другим данным, так как *TimeManager* запоминает последнее состояние времени отслеживаемой задачи.

Также при двойном щелчке мышью на элемент в списке просмотра есть возможность просмотреть информацию по выбранному элементу.

Изменение и удаление элементов происходит в окнах и формах, которые строятся в классе *EditDataFormBuilder*. Построенные им формы напоминают формы, строящиеся в классе *AddDataFormBuilder*. Там же и находится метод, который строит форму для создания отчета, в которой пользователь должен указать свои ФИО, а также организацию, если таковая имеет место быть.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.2.5. Алгоритмы и классы модуля Action

Архитектура всех без исключения плагинов для IntelliJ основывается на реализации событий и групп событий. В модуле событий (`org.taimuraztibilov.taskmanager.action`) содержатся все реализации событий плагина. Все они имеют предикат, при выполнении которого они становятся доступными для пользователя. Для удобства их использования они были сгруппированы по функциональному признаку (например, группа Add). Все события описываются в конфигурационном файле плагина *plugin.xml* (пример описания можно рассмотреть на Рисунке 3).

```
<action class="org.taimuraztibilov.taskmanager.action.TrackProjectAction"
        id="org.taimuraztibilov.taskmanager.action.TrackProjectAction" text="Choose Project To Track"
        description="Shows all existing tasks to track">
    <keyboard-shortcut first-keystroke="shift alt P" keymap="$default"/>
</action>
```

Рисунок 3. Пример объявления события в конфигурационном файле плагина *plugin.xml*

Условия доступности необходимы для корректной работы всего плагина. Так как эти условия проверяются IDE в главном потоке и с постоянной малой периодичностью, важно чтобы эти условия вычислялись быстро. Поэтому предикатные значения обновляются отдельно при вызове соответствующих событий, что является хорошим тоном при создании плагинов для платформы IntelliJ. При этом сами события обрабатываются в отдельных потоках, поэтому именно в них и происходят основные взаимодействия пользователя с плагином и плагина с IDE.

Ниже представлено сопоставление класса события с его действиями:

- *AddLabelAction*. Доступен, когда пользователь работает с проектом. Открывает форму для создания нового лейбла.
- *AddProjectAction*. Доступен, когда пользователь работает с проектом. Открывает форму для создания нового проекта, после создания автоматически назначает проект как отслеживаемый.
- *AddMilestoneAction*. Доступен, когда плагин отслеживает проект. Открывает форму для создания нового веха в текущем проекте, после создания автоматически назначает веху как отслеживаемую.
- *AddTaskAction*. Доступен, когда плагин отслеживает веху. Открывает форму для создания новой задачи, автоматического отслеживания не происходит, так как пользователь может изначально отслеживать другую задачу (например, «Создать новую веху»).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- *TrackProjectAction*. Доступен, когда пользователь работает с проектом. Открывает форму для выбора и просмотра отслеживаемого проекта.
- *TrackMilestoneAction*. Доступен, когда плагин отслеживает проект. Открывает форму для выбора и просмотра отслеживаемой вехи.
- *TrackTaskAction*. Доступен, когда плагин отслеживает веху. Открывает форму для просмотра и выбора отслеживаемой задачи, при выборе происходит проверка на наличие незакрытой контрольной точки. Если таковая имеется, то плагин предложит пользователю сохранить ключевую точку предыдущей задачи и записать время работы.
- *StopTrackingAction*. Доступен, когда плагин считает время выполнения задачи. Открывает форму для сохранения ключевой точки предыдущей задачи и записи времени работы.
- *CreateReportAction*. Доступен, когда плагин отслеживает проект. Открывает форму для указания параметров генерируемого отчета и вызывает метод менеджера отчетов.

Часть событий, отвечающих за редактирование данных в таблице реализована не была из-за нехватки времени на разработку, однако как было упомянуто ранее, все необходимые методы для редактирования данных реализованы и успешно работают в бизнес-модели плагина.

### 3.3. Обоснование выбора алгоритма решения задачи

Выбранные методы и способы взаимодействия пользователя с плагином, а также работы плагина являются достаточно быстрыми, так как сами задачи, стоящие перед разработчиком, не требуют сложных вычислений и по большей части решаются нативными решениями (Swing, SQL). Однако особое внимание в программе уделяется именно архитектуре, так как это основная сложность плагина.

### 3.4. Возможные взаимодействия программы с другими программами

Так как программа представляет собой плагин для платформы IntelliJ, она, как и другие плагины, имеет точки расширения, с помощью которых можно взаимодействовать с другими плагинами. К тому же плагин напрямую встроен в платформу, поэтому все события вызываются непосредственно самой платформой. Также платформа предоставляет различную информацию о проекте (например, его текущее местоположение), что позволяет сделать некоторые функции плагина проще.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.5. Описание и обоснование выбора метода организации входных и выходных данных

Входные данные задаются тремя способами:

- Данные получаются непосредственно из самой платформы IntelliJ посредством вызова событий с метаданными.
- Данные задаются пользователем с помощью соответствующих форм, в которых он указывает значения полей объектов.
- Данные плагин также получает из базы данных SQLite3.

Выходными данными является соответствующая запись в БД при создании или редактировании сущностей, а также в виде файла отчета формата .csv.

### 3.6. Описание и обоснование выбора состава технических и программных средств

Для корректной работы программы требуются следующие технические и программные средства:

**IDE на платформе IntelliJ**

#### **Windows OS**

- Microsoft Windows 10/8/7/Vista/2003/XP (64-bit включительно)
- RAM: 1 Гб минимум, 2 Гб рекомендовано
- Свободное место на диске: 300 Мб + не менее 1 Гб для кэша
- Минимальное разрешение экрана — 1024x768
- JDK 1.6 и выше

#### **Mac OS X**

- Mac OS X 10.5 и выше
- RAM: 1 Гб минимум, 2 Гб рекомендовано
- Свободное место на диске: 300 Мб + не менее 1 Гб для кэша
- Минимальное разрешение экрана — 1024x768

#### **Linux OS**

- GNOME или KDE
- RAM: 1 Гб минимум, 2 Гб рекомендовано
- Свободное место на диске: 300 Мб + не менее 1 Гб для кэша
- Минимальное разрешение экрана — 1024x768
- JDK 1.6 и выше

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ****4.1. Предполагаемая потребность**

Плагин может быть полезен заказчикам, желающим отслеживать процесс разработки ПО исполнителем. Также проект может заинтересовать отдельных разработчиков для мониторинга собственной продуктивности.

Плагин можно развивать – база данных разрабатывалась с учетом возможного расширения функционала в будущем (в частности, можно интегрировать поддержку GitLab, GitHub и Trello).

**4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами**

На момент создания приложения в сети не было обнаружено других плагинов, позволяющих мониторить процесс разработки и выполнения задач, а также генерации отчетов, однако наиболее близким приложением для этого является YouTrack.

Среди аналогов можно отметить Trello, которая позволяет управлять задачами в веб-приложении, GitLab и GitHub, которые содержат в себе функциональность, доступную в моем плагине и позволяют просто и эффективно отслеживать выполнение задач и создавать пулл-реквесты, а также упомянутая ранее YouTrack.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

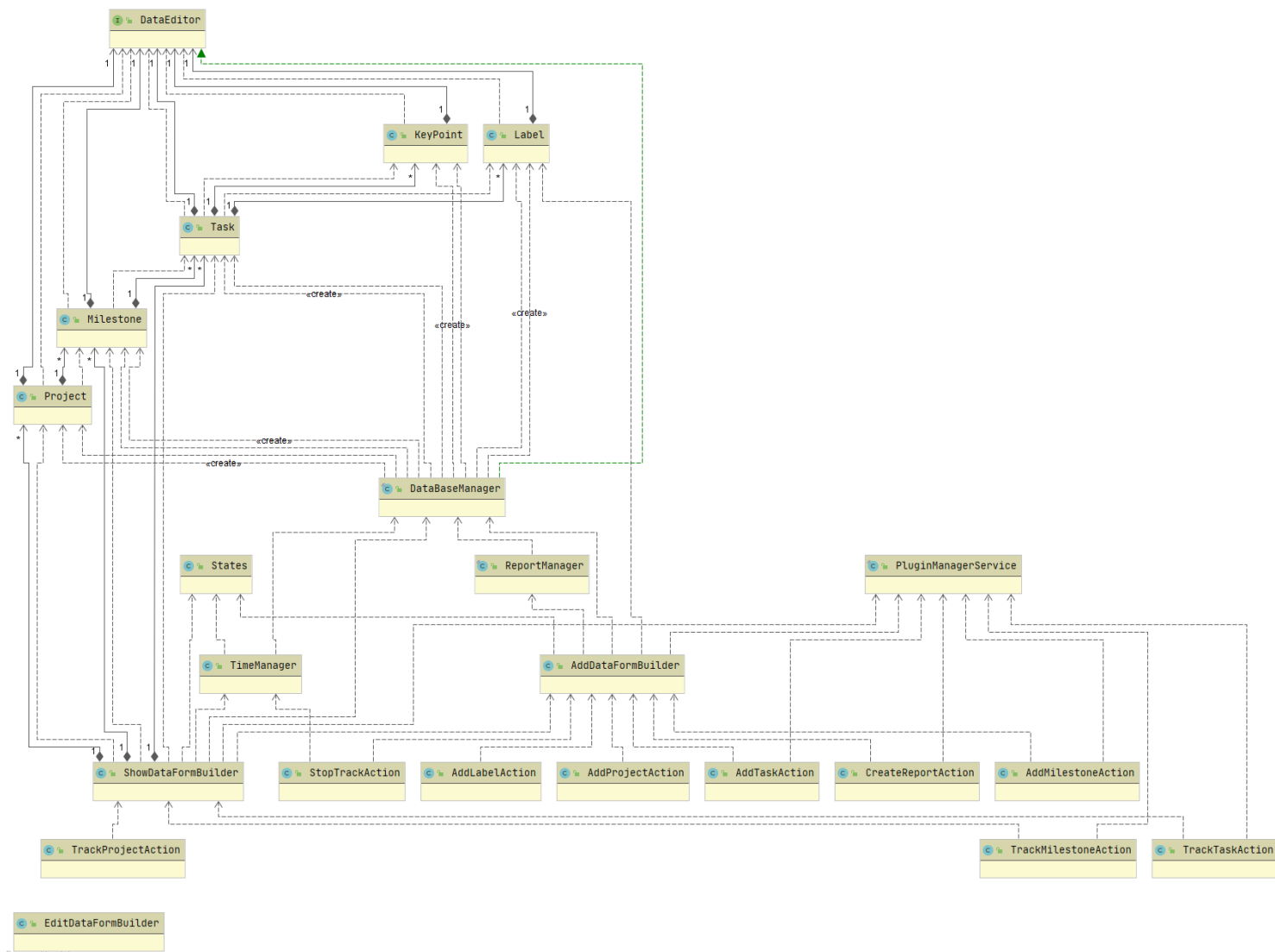
**5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ**

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. Википедия – свободная энциклопедия [Электронный ресурс] // URL: [https://ru.wikipedia.org/wiki/Заглавная\\_страница](https://ru.wikipedia.org/wiki/Заглавная_страница) (Дата обращения 29.11.19, режим доступа: свободный).
11. IntelliJ IDEA Platform SDK Guide [Электронный ресурс] // URL: <http://www.jetbrains.org/intellij/sdk/docs/welcome.html> (Дата обращения 29.11.19, режим доступа: свободный).
12. Java Platform Standard Edition 8 Documentation [Электронный ресурс] // URL: <https://docs.oracle.com/javase/8/docs/> (Дата обращения 29.11.19, режим доступа: свободный).
13. IntelliJ Platform UI Guidelines [Электронный ресурс] // URL: <https://jetbrains.design/intellij/> (Дата обращения 20.04.2020, режим доступа: свободный).
14. Open API IntelliJ Repository [Электронный ресурс] // URL: <https://upsources.jetbrains.com/idea-ce/structure/idea-ce-40e5005d02df57f58ac2d498867446c43d61101f/platform/editor-ui-api> (Дата обращения 20.04.2020, режим доступа: свободный).
15. SQLite3 documentation (with JDBC) [Электронный ресурс] // URL: <https://www.sqlitetutorial.net/> (Дата обращения 20.04.2020, режим доступа: свободный).
16. Java Swing Tutorials and Documentation [Электронный ресурс] // URL: <http://java-online.ru/libs-swing.xhtml> (Дата обращения 20.04.2020, режим доступа: свободный).
17. OpenCSV Documentation [Электронный ресурс] // URL: <http://opencsv.sourceforge.net/> (Дата обращения 20.04.2020, режим доступа: свободный).
18. Gradle Guideline and Documentation for Java 11 [Электронный ресурс] // URL: [https://docs.gradle.org/current/userguide/building\\_java\\_projects.html](https://docs.gradle.org/current/userguide/building_java_projects.html) (Дата обращения 20.04.2020, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

15  
 RU.17701729.04.01-01 81 01-1  
 UML-ДИАГРАММА КЛАССОВ

ПРИЛОЖЕНИЕ 1



Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ОСНОВНЫХ КОМПОНЕНТОВ

Модуль	Назначение
<i>org.taimuraztibilov.taskmanager.base</i>	Пакет, содержит основные классы бэкенда плагина
Классы модуля	Назначение
<i>Project</i>	Класс, модель проекта, содержит поля, соответствующие столбцам в базе данных, а также геттеры и сеттеры к ним
<i>Task</i>	Класс, модель задачи, содержит поля, соответствующие столбцам в базе данных, а также геттеры и сеттеры к ним
<i>KeyPoint</i>	Класс, модель ключевой точки задачи, содержит поля, соответствующие столбцам в базе данных, а также геттеры и сеттеры к ним
<i>Label</i>	Класс, модель лейбла задачи, содержит поля, соответствующие столбцам в базе данных, а также геттеры и сеттеры к ним
<i>States</i>	Класс, содержит числовое представление состояний задачи, вехи, проекта
<i>DataEditor</i>	Интерфейс, содержит абстрактные методы, позволяющие модифицировать модель данных
<i>DataBaseManager</i>	Класс, сервис-середжер для работы с данными в базе данных SQLite3
<i>TimeManager</i>	Класс, сервис-менеджер для работы с подсчетом времени, затраченного на задачу
<i>ReportManager</i>	Класс, сервис-менеджер для генерации отчетов по разработке
<i>PluginManagerService</i>	Класс, сервис-менеджер для хранения и изменения текущей рабочей информации по работе плагина
Модуль	Назначение
<i>org.taimuraztibilov.taskmanager.ui</i>	Пакет, содержит основные классы для работы с графическим интерфейсом и созданием пользовательских форм
Классы модуля	Назначение
<i>AddDataFormBuilder</i>	Класс, содержит методы для создания форм, которые будут использованы пользователем для создания новых объектов и записи сущностей в базу данных с помощью сервиса
<i>ShowDataFormBuilder</i>	Класс, содержит методы для создания форм, которые будут использованы пользователем для отображения объектов и выбора отслеживаемых задач с помощью сервиса

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

<i>EditDataFormBuilder</i>	Класс, задумывался как строитель форм для изменения данных в программе.
<b>Модуль</b>	<b>Назначение</b>
<i>org.taimuraztibilov.taskmanager.action</i>	Пакет, содержит основные классы для работы с событиями, генерируемыми API платформы IntelliJ для вызова соответствующих действий
<b>Классы модуля</b>	<b>Назначение</b>
<i>AddLabelAction</i>	Класс события, вызывает метод создания формы для добавления нового лейбла
<i>AddProjectAction</i>	Класс события, вызывает метод создания формы для добавления нового проекта
<i>AddMilestoneAction</i>	Класс события, вызывает метод создания формы для добавления новой вехи
<i>AddTaskAction</i>	Класс события, вызывает метод создания формы для добавления новой задачи
<i>CreateReportAction</i>	Класс события, вызывает метод создания формы для генерации отчета по проекту
<i>StopTrackAction</i>	Класс события, вызывает метод создания формы для создания ключевой точки задачи и записи потраченного времени
<i>TrackProjectAction</i>	Класс события, вызывает метод создания формы для выбора отслеживаемого проекта
<i>TrackMilestoneAction</i>	Класс события, вызывает метод создания формы для выбора отслеживаемой вехи
<i>TrackTaskAction</i>	Класс события, вызывает метод создания формы для выбора отслеживаемой задачи

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ТЕРМИНОЛОГИЯ

**Плагин** – независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Плагины обычно выполняются в виде библиотек общего пользования.

**Мониторинг (процесса разработки)** – возможность определения состояния процесса разработки ПО, например, с помощью периодически создаваемой исполнителем отчетности.

**API или Программный интерфейс** – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.

**Исполнитель** – физическое (юридическое) лицо, занимающееся выполнением поставленных задач и/или оказанием услуг за определенную плату.

**Заказчик** – физическое (юридическое) лицо, предоставляющее заказ исполнителю, заключает договор на его исполнение за определенную плату.

**Событие** – в API IntelliJ это способ взаимодействия плагина с платформой, описанная процедура, выполняемая плагином при выполнении некоторых условий.

**Группа событий** – в API IntelliJ группа задает единую точку вызова для событий, объединенных в эту группу.

**Синглтон** – паттерн, при использовании которого возможно создание только одного объекта на всю программу. В плагине используется для создания платформой сервисов бизнес-модели.

**Бизнес-модель** – абстрактная модель или вполне конкретная реализация взаимодействий компонентов и процессов программы между собой для получения некоторого результата.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

