

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**  
**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

**Факультет компьютерных наук**  
**Департамент программной инженерии**

**СОГЛАСОВАНО**  
Научный руководитель,  
Разработчик

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

\_\_\_\_\_ Ашурбеков З. И.  
«30» Мая 2024 г.

\_\_\_\_\_ Павлочев Н. А.  
«30» Мая 2024 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.02.06-01 01-1

**Поддержка разделения потребления по продуктам для выгрузки в**  
**Я. Баланс**  
**Пояснительная записка**  
**ЛИСТ УТВЕРЖДЕНИЯ**  
**RU.17701729.02.06-01**

**Исполнитель:**  
студент группы БПИ229

\_\_\_\_\_  
/ Трохачев А. В. /  
«30» Мая 2024 г.

УТВЕРЖДЕН  
RU.17701729.02.06-01 01-1-ЛЮ

Име. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
RU.17701729.02.06-01 01-1-ЛЮ				

Поддержка разделения потребления по продуктам для выгрузки в  
Я. Баланс  
Пояснительная записка  
RU.17701729.02.06-01 01-1-ЛЮ  
Листов 15

## СОДЕРЖАНИЕ

<b>1</b>	<b>ВВЕДЕНИЕ .....</b>	<b>6</b>
1.1	Наименование программы.....	6
<b>2</b>	<b>НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ .....</b>	<b>7</b>
2.1	Назначение программы.....	7
2.1.1	Функциональное назначение.....	7
2.1.2	Эксплуатационное назначение.....	7
2.2	Краткая характеристика области применения.....	7
<b>3</b>	<b>ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....</b>	<b>8</b>
3.1	Постановка задачи на разработку программы .....	8
3.2	Описание и обоснование архитектуры программы .....	8
3.2.1	Описание архитектуры программы.....	8
3.2.2	Обоснование архитектуры программы.....	9
3.3	Описание и обоснование алгоритма работы программы .....	11
3.3.1	Описание алгоритма работы программы .....	11
3.3.2	Обоснование алгоритма работы программы .....	12
3.4	Описание и обоснование выбора способа организации входных и выходных данных	13
3.5	Описание и обоснование выбора состава технических и программных средств	13
3.5.1	Состав технических и программных средств.....	13
<b>4</b>	<b>ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ .....</b>	<b>13</b>
4.1	Ориентировочная экономическая эффективность .....	13
4.2	Предполагаемая потребность.....	13
	<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>14</b>

## АННОТАЦИЯ

Данный программный документ представляет собой пояснительную записку к программному проекту «Поддержка разделения потребления по продуктам для выгрузки в Я. Баланс».

Раздел «Введение» включает в себя наименование программы и документ, на основании которого ведётся разработка, с указанием организации, утвердившей данный документ.

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

В разделе «Технические характеристики» присутствуют следующие подразделы: постановка задачи на разработку программы, описание функционирования программы, описание и обоснование алгоритма работы программы, описание и обоснование выбора метода организации входных и выходных данных, описание работы с базой 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.

Программный 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 следует ознакомиться с терминологией, приведенной в Приложении 1.

## 1 ВВЕДЕНИЕ

### 1.1 Наименование программы

**Наименование темы разработки:** «Поддержка разделения потребления по продуктам для выгрузки в Я. Баланс».

**Наименование темы разработки на английском языке:** «Support for Separating Product Consumption for Uploading to Ya. Balance».

**Условное обозначение темы разработки:** «Support for Separating Product Consumption for Uploading to Ya. Balance».

## **2 НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**

### **2.1 Назначение программы**

#### **2.1.1 Функциональное назначение**

Функциональное назначение решения «Поддержка разделения потребления по продуктам для выгрузки в Я. Баланс» заключается в предоставлении возможности разделения потребления клиентов Яндекс.Облака по типам предоставляемых услуг. Решение позволяет отдельно считать потребление, баланс, долг, выгружать данные в необходимом формате в Яндекс.Баланс и выставлять счета.

#### **2.1.2 Эксплуатационное назначение**

Аудиторией проекта являются клиенты Яндекс.Облака. Функционал проекта позволит оплачивать предоставленные услуги более гибко. Важно предоставить бизнес-аналитикам такие инструменты и возможности, которые позволят включать функционал на отдельных клиентах для удовлетворения специфических потребностей.

### **2.2 Краткая характеристика области применения**

"Поддержка разделения потребления по продуктам для выгрузки в Я. Баланс" - это комплекс программных решений, разрабатываемых с целью получения возможности разделять потребление и выставлять счета клиентам Яндекс.Облака в разрезе типов предоставляемых услуг.

### **3 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**

#### **3.1 Постановка задачи на разработку программы**

В результате разработки должен поддерживаться следующий функционал:

- 1) Возможность проставлять тип услугам Яндекс.Облака. Услуги без типа должны иметь значение по умолчанию для обратной совместимости.
- 2) Возможность платежному аккаунту иметь несколько балансов разного типа. Для каждого баланса платежного аккаунта потребление, баланс и долг рассчитывается отдельно. Этот функционал необходим для выставления счетов.
- 3) Возможность синхронизировать балансы платежного аккаунта между Яндекс.Облаком и системой Яндекс.Баланс. Это необходимо для отображения пользователю актуальной информации.
- 4) Возможность включать и выключать функционал для новосозданных платежных аккаунтов.
- 5) Возможность включать и выключать функционал для уже созданных платежных аккаунтов.
- 6) Возможность включать и выключать функционал для конкретного платежного аккаунта. Этот функционал необходим для крупных клиентов Яндекс.Облака.
- 7) Возможность отгружать потребление по разным типам услуг в Яндекс.Баланс для дальнейшей обработки.
- 8) Возможность получить детальный отчет о потреблении за расчетный период для каждого типа услуг.

#### **3.2 Описание и обоснование архитектуры программы**

##### **3.2.1 Описание архитектуры программы**

Для выполнения задачи пришлось разработать с нуля или модифицировать большую часть функционала проекта. В основном изменения затронули функциональность, связанную с платежной системой биллинга Яндекс.Облака. Список изменений представлен ниже:

- 1) Добавлен новый класс SkuLabel, отвечающий за хранение дополнительной информации об Sku. Этот класс используется для хранения типа продукта,



который хранится в поле `product_type`. Класс `SkuLabel` расширяет класс `Sku` и связан с последним отношением один к одному.

2) В таблице балансов платежных аккаунтов в качестве `primary key` вместо `id` платежного аккаунта теперь используется составной `id` на основе `id` платежного аккаунта пользователя и типа баланса. Изменены функции работы с балансами платежных аккаунтов для поддержки новой функциональности. Изменения обратно совместимы, так как раньше `id` баланса был равен `id` платежного аккаунта.

3) Транзакции теперь хранят информацию о типе баланса, с которым была проведена операция. Изменены функции работы с транзакциями для корректной работы прошлого функционала, опирающиеся на операции с балансом только определенного типа.

4) Синхронизация между системами Яндекс.Баланс и Яндекс.Облако теперь поддерживает балансы нескольких типов. Доработан прошлый функционал для поддержки консистентной работы системы вне зависимости от настроек системы, поскольку необходимо гарантировать равенство в цене потребленных услуг вне зависимости их разбивки на продукты.

5) Добавлены определенные флаги в конфиги окружений, позволяющие включать или выключать функциональность для всех пользователей.

6) Разработаны инструменты, позволяющие точно включать или выключать функциональность для отдельных пользователей. Эти инструменты доступны как разработчикам, так и менеджерам и бизнес-аналитикам.

7) Модифицирована функциональность закрытия месяца для поддержки отдельной выгрузки для различных типов продуктов. Внесены правки в предыдущий формат выгрузки для поддержки новых изменений. Добавлен мониторинг расхождения сумм потребления для всех клиентов Яндекс.Облака чтобы гарантировать корректный подсчет стоимости потребления.

8) Модифицирована функциональность составления отчета о потреблении пользователя для поддержки различных продуктов. Добавлена возможность генерировать сразу несколько отчетов: по каждому на каждый тип продукта.

9) Добавлена возможность корректно выставлять счета для разных типов балансов для клиентов Яндекс.Облака.

### **3.2.2 Обоснование архитектуры программы**

Каждое принятое архитектурное решение было одобрено командой разработчиков. Обоснования архитектурных решений:

1) Было принято решение добавить в проект отдельный класс в связи с избыточной нагрузкой на уже существующий класс `Sku`. Разбиение модели на несколько классов позволило избежать проблеме появления в проекте “God класса”.

- 2) Использование составного id было принято в связи с трудностями, возникающими при смене primary key у таблицы, и недостатком времени на разработку. Модификация уже существующего функционала для поддержки типа продукта была минимальная, поскольку id основного баланса не менялось. Это решение помогло сократить время, потраченное на разработку, и успеть сдать проект в срок.
- 3) Для возможности однозначно восстанавливать историю пользовательского баланса потребовалось добавление типа баланса в транзакцию. Это решение позволило сильно не менять уже существующий функционал и заложить расширение на будущее. Функционал, который опирался на операции определенного типа, был модифицирован с добавлением фильтрации по типу баланса.
- 4) Синхронизация с Яндекс.Баланс – это способ узнавать об изменениях в балансе пользователя и актуализировать их. Данный механизм используется для реагирования на пополнения баланса или на возврат средств через платежную систему. Синхронизация запускает сразу множество процессов, поэтому является одним из самых сложных мест проекта. Несмотря на то, что прошлый механизм синхронизации был уже перегружен, было принято решение о его расширении. Благодаря этому решению не пришлось переписывать уже существующие механизмы, а только доработать их и написать по аналогии новый. Решение значительно сократило сроки разработки и помогло сдать проект в срок.
- 5) В связи с тем, что на определенных окружениях функционал проекта будет всегда выключен, было принято решение о добавлении параметров в конфиг. Конфиги хранят различные настройки, часто зависящие от окружения. Это позволило легко настраивать и включать или выключать функционал в зависимости от окружения.
- 6) В связи с необходимостью точно включать функционал для уже существующих пользователей было принято решение ввести специальный feature flag, отвечающий за возможность платежного аккаунта иметь сразу несколько балансов.
- 7) Было принято решение не переписывать с нуля старую выгрузку в Яндекс.Баланс и не обобщать ее, чтобы сократить время на разработку и случайно не сломать критически важный функционал системы. Приняли решение написать новую выгрузку, с учетом различия в формате и деталях. Договорились, что необходим мониторинг в системе Яндекс.Мониторинг, который будет сверять суммы, полученные обеими выгрузками.
- 8) В связи с различиями между услугами по типу, было решено обобщить составление детального отчета о потреблении для поддержки разбиения на типы. Это поможет пользователям лучше понимать, как формируются счета.
- 9) Было принято решение обобщить механику выставления счетов для гибкого создания нескольких счетов для одного клиента.

### 3.3 Описание и обоснование алгоритма работы программы

#### 3.3.1 Описание алгоритма работы программы

Алгоритм, синхронизирующий баланс платежного аккаунта с системой Яндекс.Баланс:

1) Выяснить, включен ли функционал отдельной загрузки для синхронизируемого платежного аккаунта. Для этого необходимо получить значения параметров в конфиге и посмотреть, есть ли на платежном аккаунте feature flag. Считается, что платежный аккаунт попадает под отдельную загрузку только в том случае, если оба условия соблюдены.

2) Получить составной id баланса платежного аккаунта в зависимости от типа баланса.

3) Получить из базы данных последнее синхронизированное значение баланса для полученного составного id.

4) Получить текущее состояние баланса пользователя из системы Яндекс.Баланс.

Если значение изменилось, то создать транзакцию для сохранения истории изменения баланса платежного аккаунта с проставлением в ней нужного типа баланса. Если баланс увеличился, то транзакция считается пополнением, иначе – возвратом денежных средств через банк.

5) Рассчитать стоимость текущего потребления ресурсов, оплачиваемых этим платежным аккаунтом в разрезе типа продукта баланса.

6) Вычислить текущий размер баланса пользователя как: сумма пополнений – сумма потребления.

7) В зависимости от размера баланса платежного аккаунта предпринять действия:

Если баланс не изменился, то ничего делать не нужно.

Если баланс изменился в отрицательную сторону и стал меньше нуля, то отправить пользователю письмо о необходимости пополнения баланса.

Если баланс превысил максимально допустимое отрицательное значение, то нужно заблокировать пользователя и его сущности, оплачиваемые этим балансом за долг.

8) Обновить значение пользовательского баланса в базе данных

Алгоритм, выгружающий данные о потреблении пользователя в Яндекс.Баланс в разбивке на типы продуктов:

1) Получить Sku, подходящие для продукта. Для этого нужно выполнить inner join таблицы Sku на таблицу SkuLabels и отфильтровать результат по product\_type. Этот шаг необходим, чтобы считать потребление только по тем услугам, которые относятся к этому продукту.

2) Подсчитать потребление по полученным Sku за текущий месяц. Для этого нужно сделать join таблицы потребления на таблицу полученных Sku.

Результат отфильтровать по дате, она должна лежать в текущем месяце.

3) Подготовить строки для выгрузки в Яндекс.Баланс на основе полученных данных. Для этого необходимо заполнить дополнительные поля и сгенерировать уникальные идентификаторы для каждой записи.

4) Выгрузить обогащенные данные в Яндекс.Баланс.

5) Убедиться, что суммарное потребление пользователя за месяц равно сумме потреблений по всем его продуктам. Если это не так, завершиться с ошибкой.

Алгоритм, составляющий отчет о потреблении пользователя:

1) Получить флаг, отвечающий за то, включен или выключен функционал отдельной выгрузки для платежного аккаунта пользователя. Для этого нужно получить значения параметров в конфиге и получить значение feature flag на платежном аккаунте пользователя. Если платежный аккаунт попадает под действия нового функционала только в том случае, если оба условия соблюдены.

2) Получить общее потребление пользователя за период. Для этого нужно сходить в базу данных и получить отфильтрованные по дате строки потребления из таблицы.

3) Отфильтровать полученные данные о потреблении по Sku, если функционал включен для этого платежного аккаунта. Для этого нужно для каждой Sku, по которой было потребление, сопоставить тип ее продукта.

4) Обогатить полученные данные информацией о сервисах, облаках и переводами.

5) Вернуть сгенерированный на основе данных pdf.

### **3.3.2 Обоснование алгоритма работы программы**

Обобщенный алгоритм синхронизации был разработан с учетом уже существовавшего алгоритма синхронизации, работающего без разделения на продукты.

Большая часть алгоритма совпадает, но существуют различия, которые не позволяют слить их воедино.

К сожалению, принятое решение о расширении уже существующего алгоритма увеличило его сложность, но позволило закончить проект в срок с минимальными рисками.

Правильность гарантирована большим тестовым покрытием как старого, так и нового функционала.

Реализация алгоритма выгрузки по продуктам значительно отличается общей реализацией. Это связано с разными форматами выгрузки и различиями между версиями, поскольку старая выгрузка писалась в 2029 году.

Сам алгоритм концептуально не менялся.

Правильность гарантирована тестами и мониторингами системы, которые в случае обнаружения расхождения призвут дежурного для анализа.

Реализация алгоритма составления отчета о потреблении пользователя была дополнена для корректной работы с новым функционалом. Корректность гарантирована тестами.

### **3.4 Описание и обоснование выбора способа организации входных и выходных данных**

Формат входных данных - метрики потребления ресурсов клиентом Яндекс.Облака.

### **3.5 Описание и обоснование выбора состава технических и программных средств**

#### **3.5.1 Состав технических и программных средств**

Изменения внесенные разработкой проекта не предусматривают изменения состава технических программных средств. Должны быть использованы стандартные средства проекта.

## **4 ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

### **4.1 Ориентировочная экономическая эффективность**

Ожидается, что проект лучшим образом скажется на гибкости системы тарификации Яндекс.Облака и улучшит пользовательский опыт, в результате чего возрастут экономические показатели.

### **4.2 Предполагаемая потребность**

Поскольку новый функционал делает систему тарификации более гибкой, ожидается значительный спрос на него у крупных клиентов Яндекс.Облака.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 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) ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

- 11) ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
- 12) ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]