SRS

Software Requirements Specification

Спецификация требований к ПО

табл. История изменения версий.

| Верс | Автор | Утвердил | Дата | Описание |
| --- | --- | --- | --- | --- |
| 1 | Шумейко А. М. | Шумейко А. М. | 25.01.2015 | Создал документ. |
|  |  |  |  |  |

Оглавление

[1 Введение 4](#_Toc444353630)

[1.1 Цель 4](#_Toc444353631)

[1.2 Область применения 4](#_Toc444353632)

[1.3 Определения, термины и сокращения 4](#_Toc444353633)

[1.4 Ссылки 4](#_Toc444353634)

[2 Общее описание 4](#_Toc444353635)

[2.1 Перспективы продукта 4](#_Toc444353636)

[2.2 Функции продукта 10](#_Toc444353637)

[2.3 Пользовательские характеристики 19](#_Toc444353638)

[2.4 Ограничения 19](#_Toc444353639)

[2.5 Предположения и зависимости 19](#_Toc444353640)

[2.6 Распределение требований 19](#_Toc444353641)

[3 Детальные требования 20](#_Toc444353642)

[3.1 Требования к внешнему интерфейсу 20](#_Toc444353643)

[3.2 Классы/Объекты 20](#_Toc444353644)

[3.3 Требования баз данных 27](#_Toc444353645)

[3.4 Требования к производительности 28](#_Toc444353646)

[3.5 Ограничения проектирования 28](#_Toc444353647)

[3.6 Атрибуты программной системы 28](#_Toc444353648)

[3.7 Дополнительные требования 28](#_Toc444353649)

[4 Сопровождающая информация 28](#_Toc444353650)

# Введение

## Цель

Этот документ предоставляет все требования для программы «Наш бюджет». Части 1 и 2 составлены с точки зрения пользователей программы, часть 3 сформулирована для разработчиков.

## Область применения

Этот документ охватывает требования к версии программы 1.0. Так же документ содержит замечания, которые могут быть применены в будущих версиях программы.

## Определения, термины и сокращения

* ОС – операционная система.
* Расход – покупки, оплата коммунальных платежей, расходы на развлечения и т. п.
* Деталь – часть расхода, расход может состоять из нескольких деталей.
* Тип детали – описание свойств детали.

## Ссылки

[Ссылки на связанные документы]

# Общее описание

Программа мой бюджет предоставит пользователям возможность управления личными и семейными средствами. У пользователей будет возможность осуществлять записи о покупках, указывая, что когда и по какой цене было куплено. По желанию пользователь может присоединять (ся) других пользователей для совместного ведения записей о покупках. Будет возможность планировать покупки, в том числе и для присоединенных пользователей. Так же будет возможность синхронизировать записи на различных устройствах. Благодаря группировке расходов по типам и сортировки их по датам будет возможно отслеживать и оптимизировать свои расходы.

## Перспективы продукта

[Здесь нужно провести обзор аналогов конкурирующих программ, это хороший способ представить перспективы продукта]

### Системные интерфейсы

Нет. Непонятно что это.

### Пользовательские интерфейсы

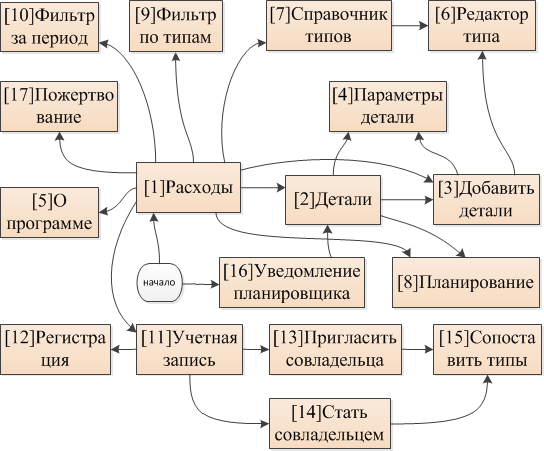


Рисунок 2.1‑1. Карта экранов приложения.

Всего в первой версии программы будет около 17 экранов (окон), в том числе и диалоговых, не считая всплывающих сообщений. На рис. Рисунок 2.1‑1 изображена карта экранов, стрелками обозначена возможность перехода от одного экрана к другому.

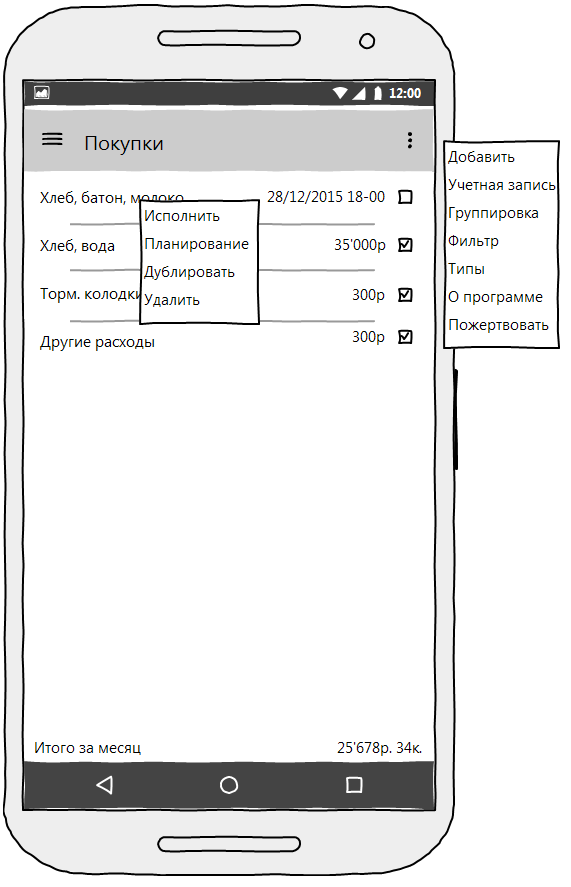


Рисунок 2.1‑4. Список покупок.

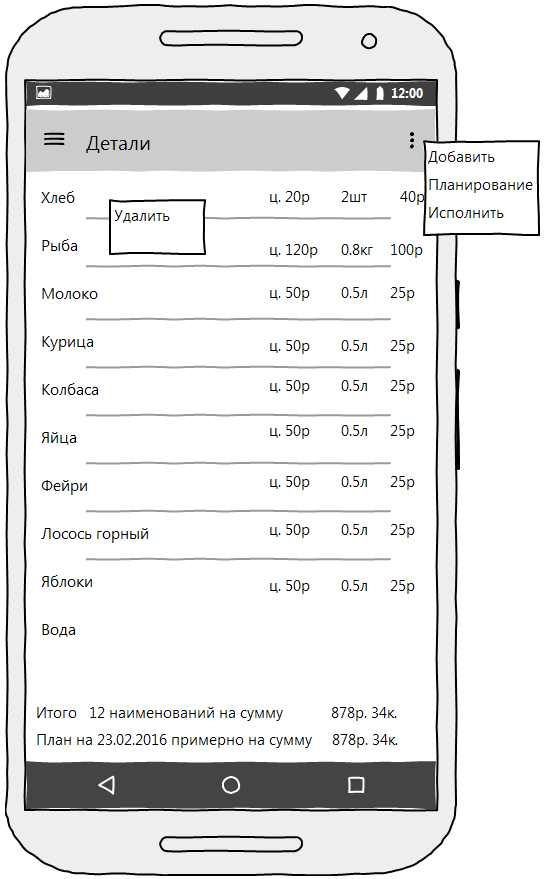


Рисунок 2.1‑4. Список деталей покупки.

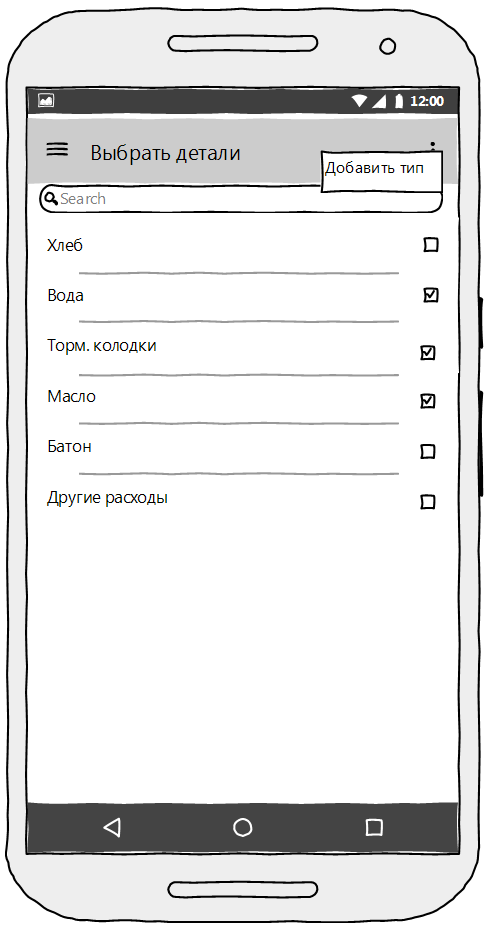


Рисунок 2.1‑4. Удобный список типов для массового выбора.

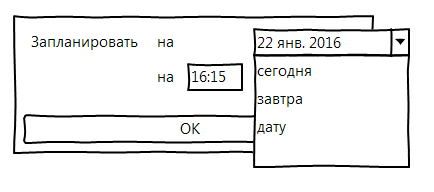


Рисунок 2.1‑7. Диалог параметров детали.

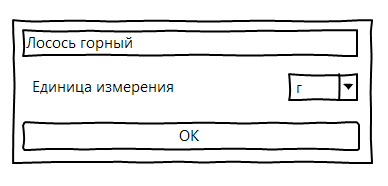


Рисунок 2.1‑7. Диалог редактора типа.

Рисунок 2.1‑7. Диалог планирования.

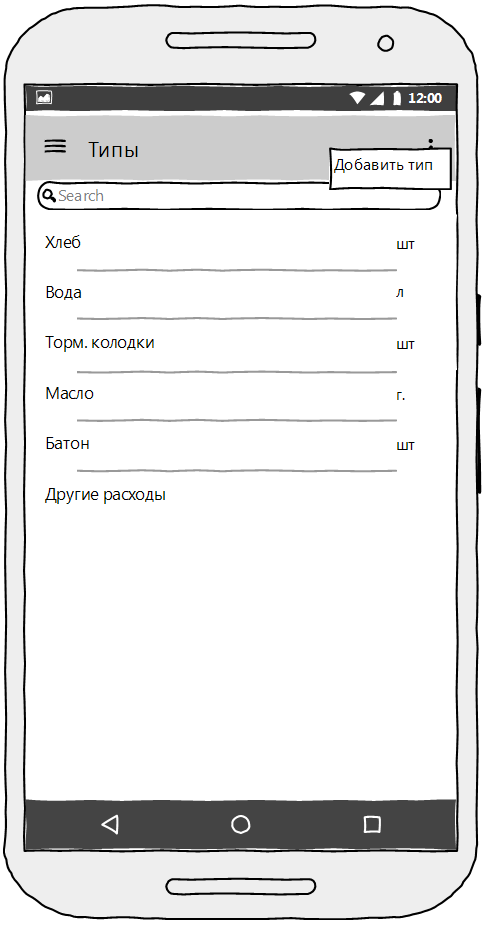
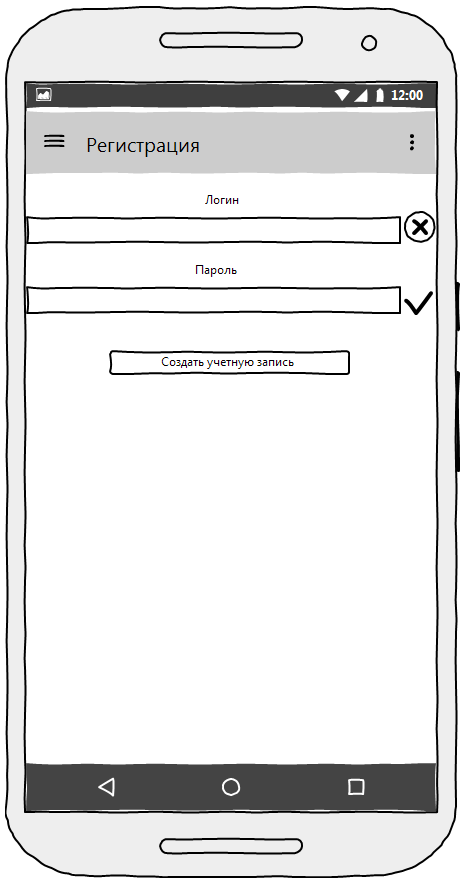


Рисунок 2.1‑10. Справочник типов.

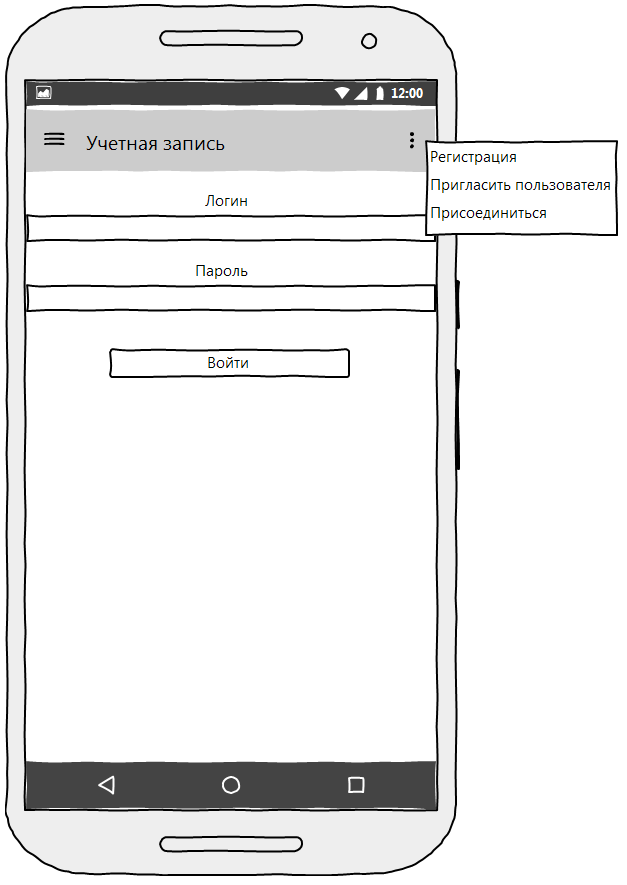


Рисунок 2.1‑10. Окно учетных записей.

Рисунок 2.1‑10. Окно рекистрации.

### Аппаратные интерфейсы

Нет.

### Программные интерфейсы

Нет.

### Коммуникационные интерфейсы

Для обмена данными между программой и сервером будет применен протокол HTTP с использованием структур данных JSON.

### Ограничения по памяти

Пока не понятно как тут, что то оценить.

### Операции

[Обычные и особенные операции требуемые от пользователя]

Может надо указать где взять и как установить программу !?

Все что может и должен делать пользователь перечислено в п. 2.2.

### Требования по адаптации

В будущих версиях программа должна стать многоязычной.

## Функции продукта

На диаграмме вариантов использования Диаграмма 2.2‑1 изображены общие действия и роли присущие программе. С точки зрения пользователя программа «Наш бюджет» должна обладать следующими возможностями:

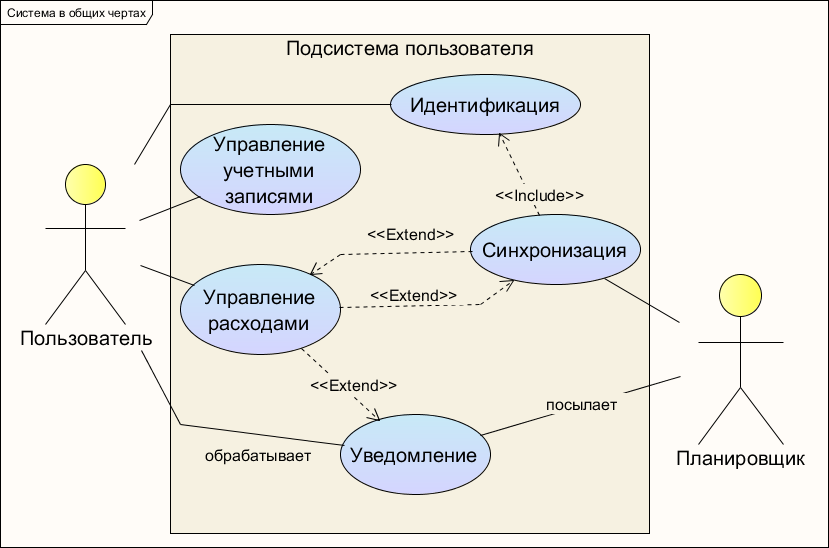


Диаграмма 2.2‑1. Общее описание системы.

### Управление учетными записями

На диаграмме вариантов использования Диаграмма 2.2‑2 более детально рассмотрена функциональность «управление учетными записями», изображенная на диаграмме Диаграмма 2.2‑1.

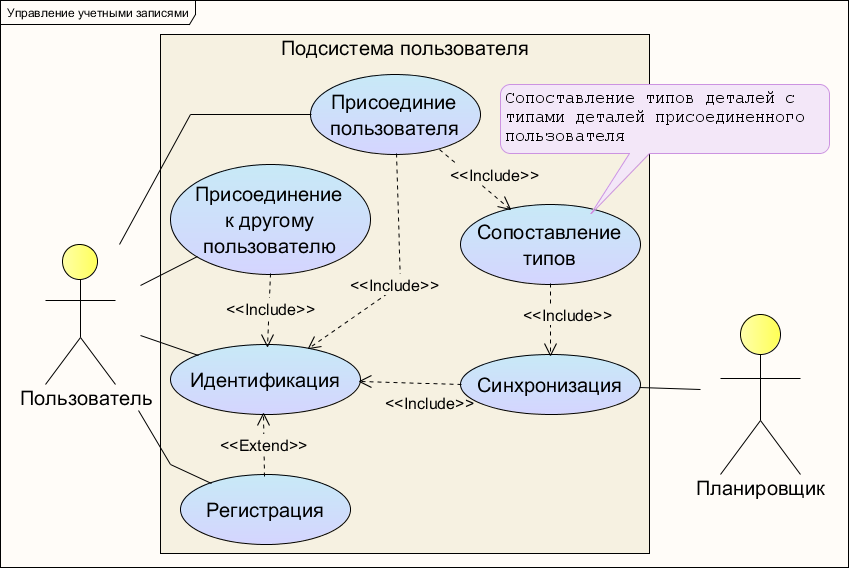


Диаграмма 2.2‑2. Управление учетными записями.

#### Должна быть возможность регистрации пользователя.

Пользователь должен иметь возможность создать учетную запись, которая состоит из имени пользователя и пароля (см. рис. Рисунок 2.1‑1 окно[12]). Создавать учетную запись можно только, если есть подключение к интернету.

#### Должна быть возможность идентификации пользователя.

Если у пользователя есть учетная запись, то он должен иметь возможность ввести идентификационные данные (см. рис. Рисунок 2.1‑1 окно[11]). Эти данные будут использованы для синхронизации и многопользовательского управления расходами.

#### Должна быть возможность присоединить пользователя к управлению расходами.

Если пользователь хочет управлять расходами совместно с другими членами своей семьи, то он должен зарегистрировать учетную запись, и перейти в режим ожидания присоединения другого пользователя (см. рис. Рисунок 2.1‑1 окно[13]), пользователь который хочет присоединиться должен выполнить действия из п. 2.2.1.4.

После того как пользователь присоединился к управлению расходами, все изменения внесенные одним пользователем должны попадать к другому пользователю через механизм синхронизации.

##### Должна быть возможность сопоставить типы деталей расходов двух пользователей.

После того как второй пользователь присоединён, необходимо сопоставить типы деталей расходов одного пользователя с типами деталей расходов другого пользователя (см. рис. Рисунок 2.1‑1 окно[15]). Например, у одного пользователя есть тип «хлеб», а у другого пользователя есть тип «хлеб черный», по факту эти типы означают одно и то же, поэтому пользователь должен указать, что эти типы означают одно и то же.

После сопоставления типов нужно произвести слияние расходов одного пользователя с расходами другого пользователя. В итоге у обоих пользователей будет полный набор расходов обоих пользователей.

#### Должна быть возможность присоединиться к другому пользователю для управления расходами.

Если пользователь хочет присоединиться к управлению расходами совместно с другими членами своей семьи, то он должен зарегистрировать учетную запись, и перейти в режим присоединения к другому пользователю (см. рис. Рисунок 2.1‑1 окно[14]). Присоединиться к другому пользователю можно только один раз.

### Управление расходами

На диаграмме вариантов использования Диаграмма 2.2‑3 более детально рассмотрена функциональность «Управление расходами», изображенная на диаграмме Диаграмма 2.2‑1.

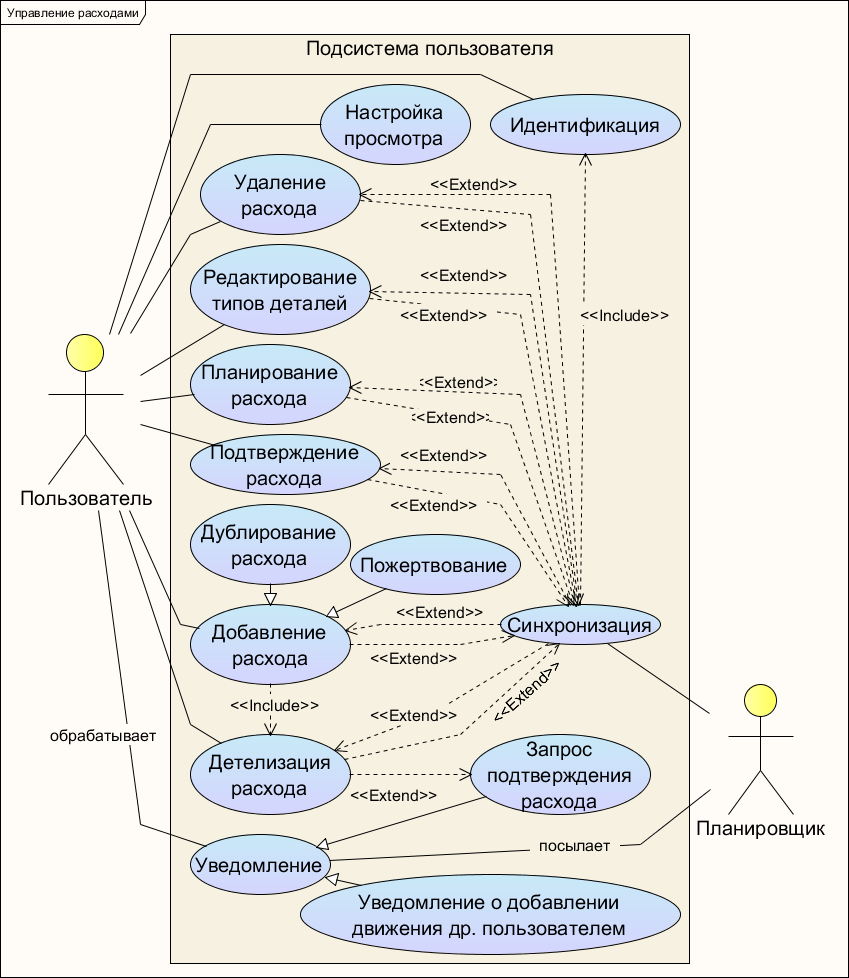


Диаграмма 2.2‑3. Управление расходами.

При запуске программы нужно отобразить список расходов (см. рис. Рисунок 2.1‑1 окно[1]).

#### Должна быть возможность настраивать просмотр списка расходов различными способами.

##### Группировка.

Группировка должна настраиваться из меню окна расходов.

###### Должна быть возможность отображать список расходов, не группируя их, то есть каждый расход отдельно.

Для совершенных расходов вывести краткий список его деталей, дату и время совершения, общую сумму всех стоимостей его деталей. Для запланированных расходов вывести краткий список его деталей, запланированную дату и время, примерную общую сумму всех стоимостей его деталей. Этот способ отображения задается по умолчанию.

###### Должна быть возможность группировать детали.

Нужно отобразить список деталей по всем совершенным (кроме запланированных) расходам, причем объединить детали с одинаковым типом, просуммировав их количество и стоимости (см. п. 2.2.3.1.1).

##### Фильтрация

###### Должна быть возможность отфильтровать список за период времени.

Нужно иметь возможность отображать список за последнюю неделю, за последний месяц, за определенный период (см. рис. Рисунок 2.1‑1 окно[10]).

###### Должна быть возможность отфильтровать список по типам детали.

Фильтрация по типам доступна только, если включена группировка по деталям (см. п. 2.2.2.1.1.2). Нужно предоставить список всех типов деталей, который сформировался после включения группировки, что бы пользователь смог отметить те типы, детали которых должны отображаться (см. рис. Рисунок 2.1‑1 окно[9]).

#### Должна быть возможность добавить расход.

Пользователь должен иметь возможность добавить новый расход, при этом нужно автоматически перейти к удобному выбору списка деталей (см. п. 2.2.3.3).

##### Должна быть возможность дублировать расход.

Нужно сделать копию расхода на основе существующего расхода, сделать такой же набор деталей, а затем перейти к детализации этого расхода (см. п. 2.2.3.2).

##### Должна быть возможность сделать пожертвование разработчику.

Пользователь должен иметь возможность пожертвовать разработчику на сопровождение и развитие данной программы (см. рис. Рисунок 2.1‑1 окно[9]). Нужно ли как, то это отображать в расходах пока не понятно, а то пользователь второй раз не пожертвует, скажет – хватит с них. Наверное, нужно сделать так, что бы иногда планировался расход – пожертвование разработчику.

#### Должна быть возможность детализировать расход.

Если выбрать расход из списка расходов, то нужно перейти к детализации расхода (см. п. 2.2.3.2).

#### Должна быть возможность удалить расход.

Нужно спросить у пользователя подтверждение удаления расхода. При удалении расхода удалить все детали по расходу.

#### Должна быть возможность отредактировать типы деталей расходов.

См. п. 2.2.3.3.1.

### Детализация расхода.

На диаграмме вариантов использования Диаграмма 2.2‑4 более детально рассмотрена функциональность «Детализация расхода», изображенная на диаграмме Диаграмма 2.2‑1.

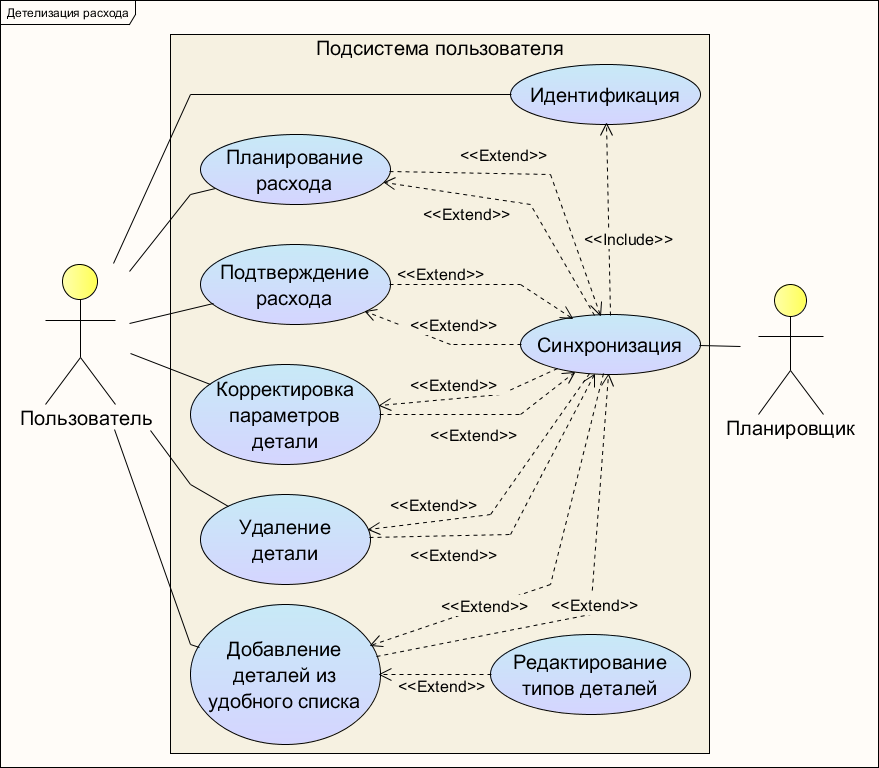


Диаграмма 2.2‑4. Детализация расхода.

#### Корректировка параметров детали.

Должна быть возможность изменить для детали единицу измерения цены и единицу измерения количества (см. п. 2.2.3.1.1), (см. рис. Рисунок 2.1‑1 окно[5]). Также должна быть возможность задать или изменить для детали цену, количество и стоимость, причем при вводе только двух параметров, третий нужно вычислить, задать можно все или только один параметр или не задавать ни одного (см. рис. Рисунок 2.1‑1 окно[4]).

##### Детали должны иметь тип, единицы измерения, цену, количество и стоимость.

Каждая деталь по расходу должна иметь тип, например тип «продукты», или тип «бензин АИ 92» (см. п. 2.2.3.3.1).

Детали должны иметь единицы измерения цены, например, цена 125руб. за 100 г-грамм, и единицы измерения количества, например вес 1,250Кг. Единицы измерения детали должны соответствовать единице измерения заданной для типа детали, допускается отличие только в десятичной приставке, например для типа «лосось горный» задана единица измерения г-граммы, тогда единицами измерения цены и единицами измерения количества могут быть как г-граммы, так и Кг-килограммы.

#### Отображение списка деталей.

При переходе к детализации расхода необходимо отобразить список деталей по данному расходу, нужно отобразить название, цену, количество, и стоимость, если они заданы (см. рис. Рисунок 2.1‑1 окно[2]). Список деталей не может быть пустым.

#### Должна быть возможность удобного и быстрого выбора списка деталей для расхода.

Необходимо вывести список типов с возможностью устанавливать флажок (см. рис. Рисунок 2.1‑1 окно[3]), так же должна быть возможность задать параметры добавляемой детали (см. п. 2.2.3.1), изначальные значения параметров необходимо брать из детали последнего расхода с таким типом. Данные по используемым единицам измерения должны быть заданы по умолчанию в любом случае, если не из детали последнего расхода, то из типа. Данный список нужно отсортировать по популярности и алфавиту. Должна быть строка поиска, которая динамически отсеивает лишние элементы этого списка при вводе в нее текста. Если искомого типа не оказалось, то должна быть возможность добавить нужный тип (см. п. 2.2.3.3.1). После того как отмечены нужные типы нужно сформировать соответствующий список деталей и перейти к отображению этого списка (см. п. 2.2.3.2). Формировать пустой список деталей запрещено.

##### Должна быть возможность редактировать типы деталей.

Типы деталей должны иметь имя и используемые единицы измерения (см. рис. Рисунок 2.1‑1 окно[6]). Список доступных единиц измерения приведен в табл. Таблица 2.2‑1.

Таблица ‑. Используемые единицы измерения.

|  |  |  |
| --- | --- | --- |
| Группа | Название | Комментарий |
| Меры количества | шт | штуки |
| дес. | десятки |
| Меры массы | г | граммы |
| Кг | килограммы = 1000гр |
| Меры объема | л | литры |
| мл | миллилитры = 0,001л |
| Меры длины | м | метры |
| см | сантиметры = 0.01м |
| Меры электроэнергии | КВт.ч | киловатты в час |
| Др. меры объема | куб.м | кубометры |
| Др. меры массы | т | тонны |

Должна быть возможность добавлять, редактировать и удалять типы деталей (см. рис. Рисунок 2.1‑1 окно[7]). После удаления данный тип нужно продолжать отображать для уже проведенных деталей, новые детали с удаленным типом добавлять нельзя. При добавлении типа нужно проверить, что название типа не дублируется.

#### Должна быть возможность запланировать расход.

Должна быть возможность указать дату и время напоминания о необходимости исполнить расход (см. рис. Рисунок 2.1‑1 окно[8]).

#### Должна быть возможность подтвердить расход.

Должна быть возможность исполнить только что добавленный или запланированный расход. При этом должна быть возможность отредактировать дату и время расхода, по умолчанию выводить текущую дату и время.

### Пользователь должен получать уведомления о событиях.

Для событий, описанных в п. 2.2.4.1.1 и п. 2.2.4.1.2 выводить одно и то же окно (см. рис. Рисунок 2.1‑1 окно[16]) с разным текстом описания события.

##### Пользователь должен получать уведомление о планировании расхода другим пользователем.

Если другой пользователь запланировал (см. п. 2.2.3.4) расход, то после синхронизации (см. п. 2.2.5) пользователю нужно выдать сообщение о том, что в списке расходов появился запланированный расход. Из этого сообщения пользователь должен мочь перейти к просмотру деталей расхода (см. п. 2.2.3.2).

##### Пользователь должен получать напоминание о запланированном расходе.

Не зависимо от того, запущена программа или нет, пользователь должен получать уведомления о запланированном расходе, если настало время, указанное при планировании (см. п. 2.2.3.4). Из этого уведомления пользователь должен мочь перейти к просмотру деталей расхода (см. п. 2.2.3.2).

### Синхронизация.

Если пользователь идентифицировался, то не зависимо от того, запущена программа или нет, должна происходить синхронизация измененных данных внесенных всеми пользователями. После синхронизации набор данных должен быть одинаковым у всех пользователей.

При использовании одних и тех же идентификационных данных, так же должна происходить синхронизация на всех устройствах, где используются эти идентификационные данные.

## Пользовательские характеристики

Предполагается, что данной программой сможет пользоваться широчайший круг пользователей с возрастом от 14лет.

## Ограничения

1. Программа будет работать на устройствах под управлением ОС Android 4.0 и на более поздних версиях. Языком разработки будет Java. Локальная СУБД будет встроенная в ОС Android СУБД SQLite.
2. Сервер синхронизации будет работать на арендованном интернет хостинге, под управлением HTTP-сервера Apache и интерпретатором PHP, используемый язык разработки PHP 5.x, хранение данных будет осуществляться под управлением СУБД MySql x.x.

## Предположения и зависимости

Нет.

## Распределение требований

[Порядок, в котором требования будут выполняться.]

Потом заполнить примерно как в книге.

# Детальные требования

## Требования к внешнему интерфейсу

### Пользовательские интерфейсы

### Аппаратные интерфейсы

Нет.

### Программные интерфейсы

### Коммуникационные интерфейсы

Описать прикладной протокол обмена данными клиента с сервером, возможно при помощи диаграммы Вирта.

## Классы/Объекты

### Диаграммы классов

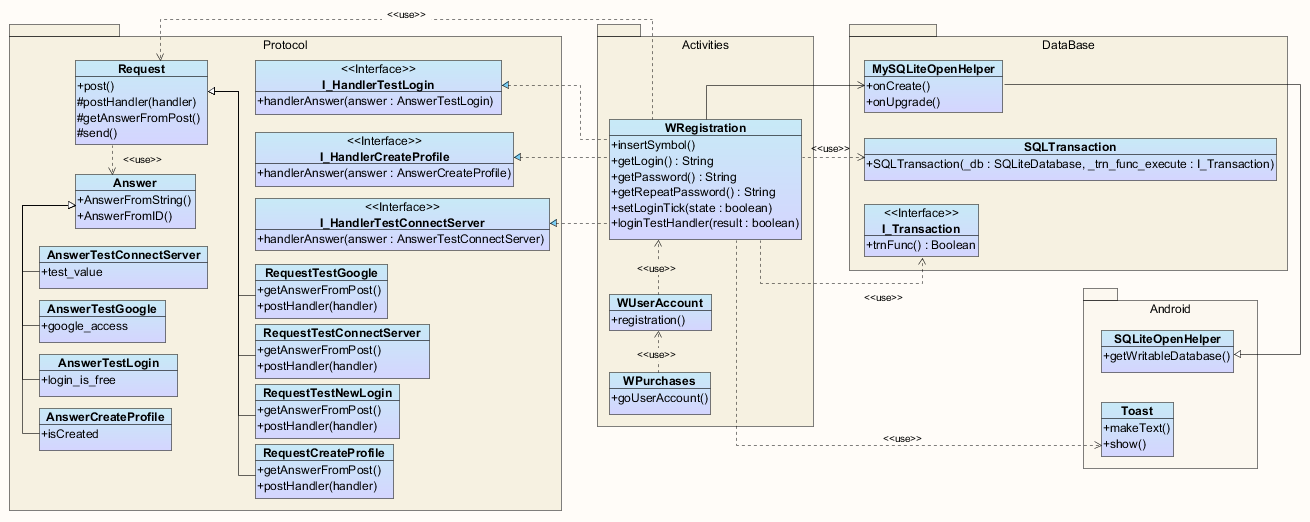


Диаграмма ‑. Диаграмма классов задействованных в функционале окна регистрации.

### Диаграммы последовательности и деятельности

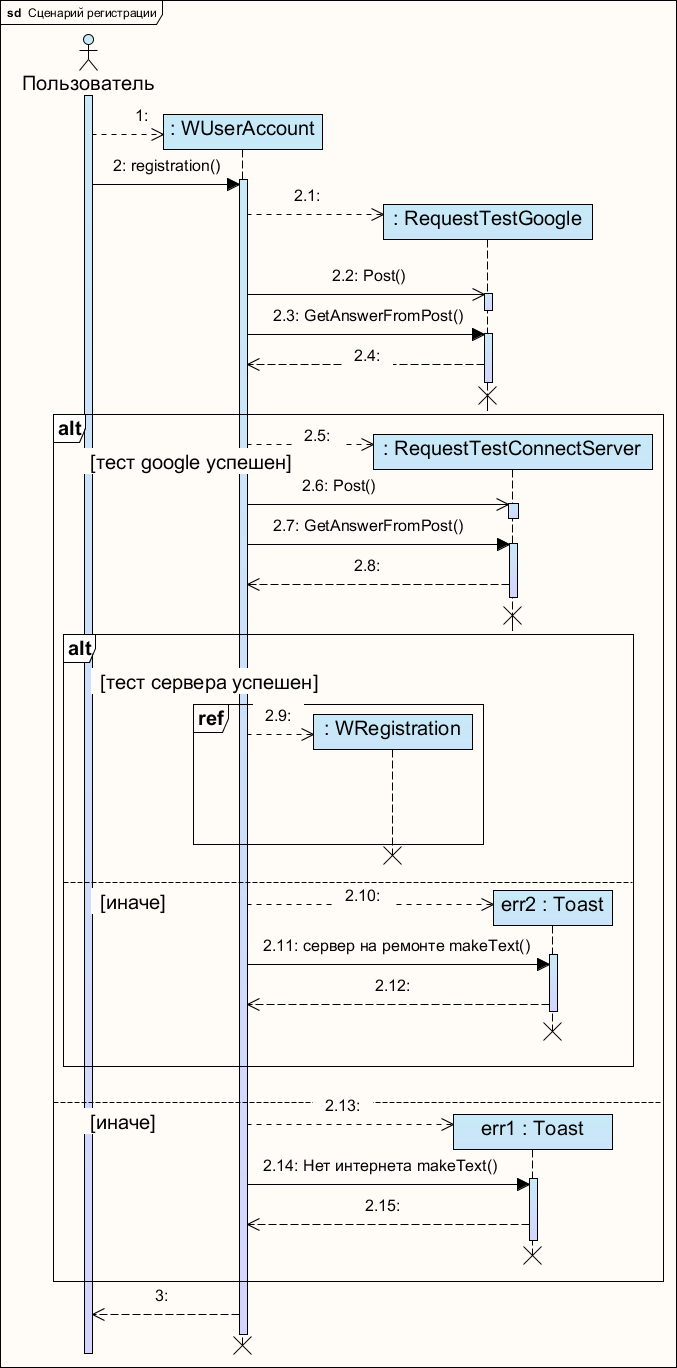


Диаграмма ‑. Сценарий перехода из окна учетных записей в окно регистрации.

#### Сценарий перехода к регистрации новой учетной записи пользователя.

1. В окне учетных записей пользователя выбираем пункт меню Регистрация.
2. Проверяем доступность адреса google.com, если недоступен, то выводим сообщение «Отсутствует подключение к интернет» и выходим из алгоритма.
3. Проверяем доступность сервера синхронизации, если недоступен, то выводим сообщение «Сервер временно недоступен» и выходим из алгоритма.
4. Переходим в окно регистрации.

Данный сценарий изображен на диаграмме Диаграмма 3.2‑2.

#### Сценарий регистрации новой учетной записи.

Пользователь ввел очередной символ логина.

1. Если длина логина >= 3, то переходим к п. 3, иначе переходим к п. 4.
2. Проверяем на сервере, не занят ли такой логин, если занят, то переходим к п. 4. иначе переходим к п. 3.
3. Выставляем разрешающую иконку и выходим из алгоритма.
4. Выставляем запрещающую иконку и выходим из алгоритма.
5. Пользователь ввел очередной символ пароля.

Пользователь ввел очередной символ пароля

1. Если длина пароля >=4, то выставляем разрешающую иконку, иначе выставляем запрещающую иконку.

Пользователь нажал на кнопку регистрация.

1. Проверить логин в соответствии с алгоритмом обработки ввода очередного символа логина.
2. Если проверка успешна, то переходим к п. 3, иначе выводим сообщение «Введите другой логин» и выходим из алгоритма.
3. Проверить пароль в соответствии с алгоритмом обработки ввода очередного символа пароля.
4. Если проверка успешна, то переходим к п. 5, иначе выводим сообщение «Введите другой пароль» и выходим из алгоритма.
5. Проводим регистрацию нового пользователя на сервере и возвращаемся в предыдущее окно учетных записей пользователя.

Данный сценарий изображен на диаграмме Диаграмма 2.2‑3 и диаграмме Диаграмма 3.2‑4.

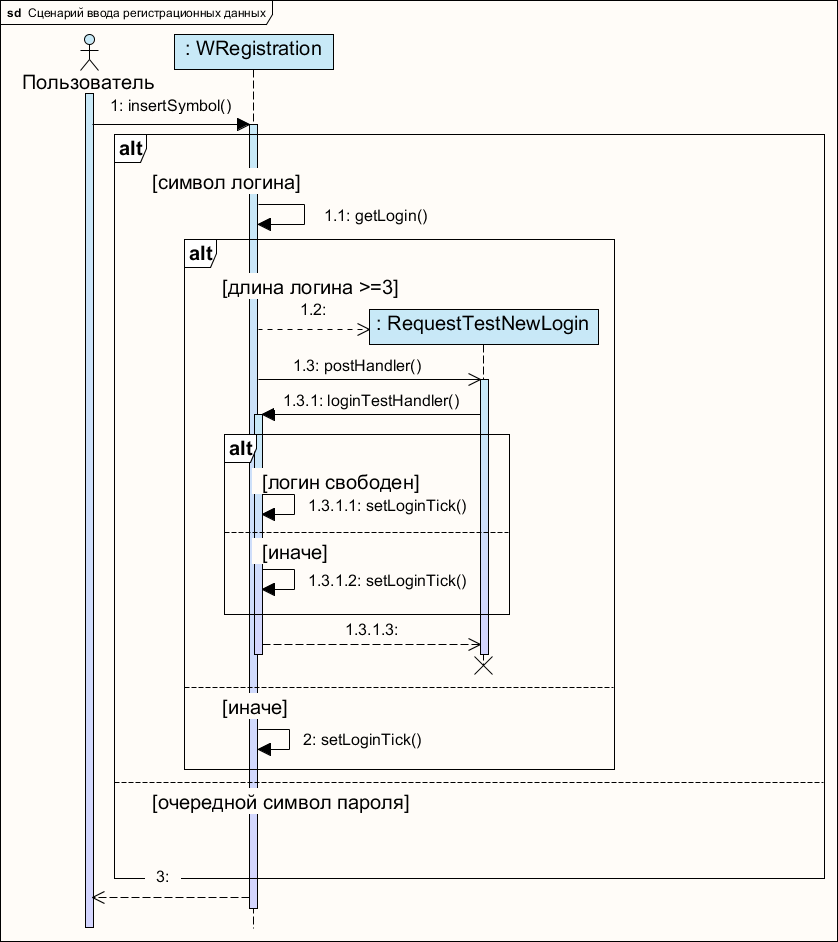


Диаграмма ‑. Сценарий регистрации учетной записи.

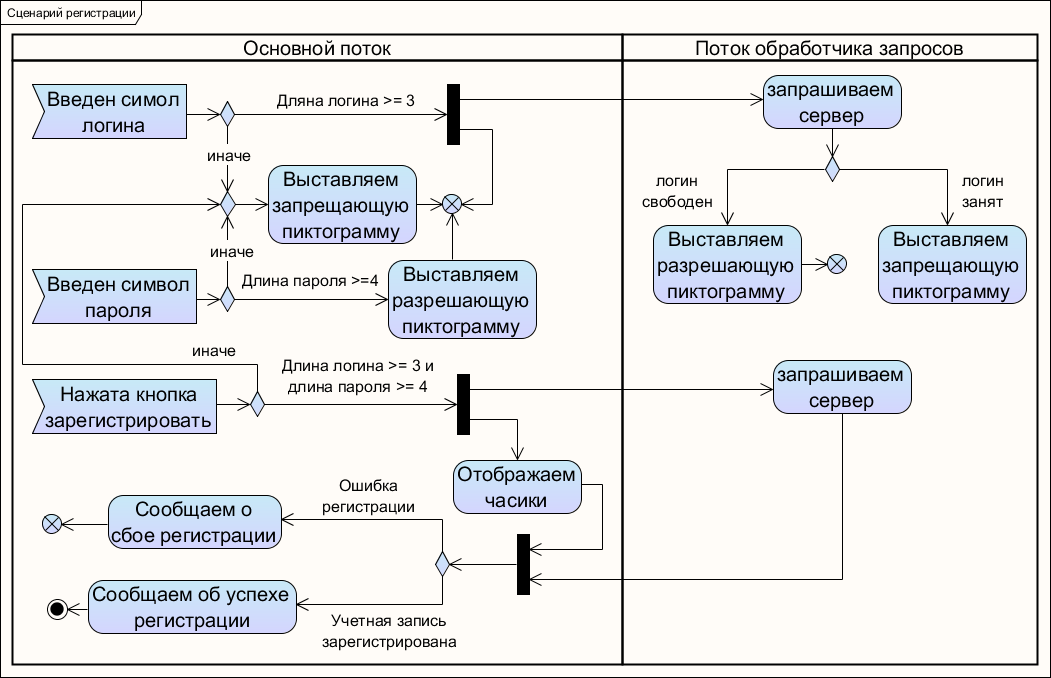


Диаграмма ‑. Сценарий регистрации учетной записи.

## Требования баз данных

### Требования БД клиентской подсистемы

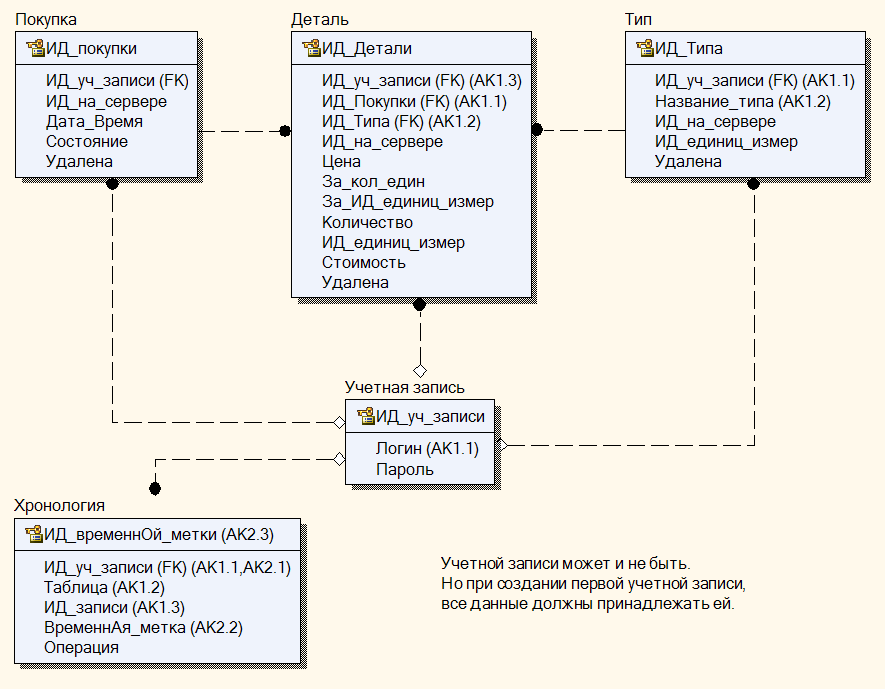


Рисунок 3.3‑1. Диаграмма БД сущность связь подсистемы клиента.

Все индексы добавлены для уникальности записей, а составной индекс таблицы «Хронология» «AK2.x» создан для ускорения поиска записей по полям «ИД\_уч\_записи» и «ВременнАя\_метка», третье поле «ИД\_временнОй\_метки» добавлено потому, что СУБД SQLite поддерживает только уникальные индексы, а индекс «AK2.x» (AK – alternate key) должен быть дублируемым, решается вводом в индекс уникального поля последним сегментом индекса.

### Требования БД серверной подсистемы

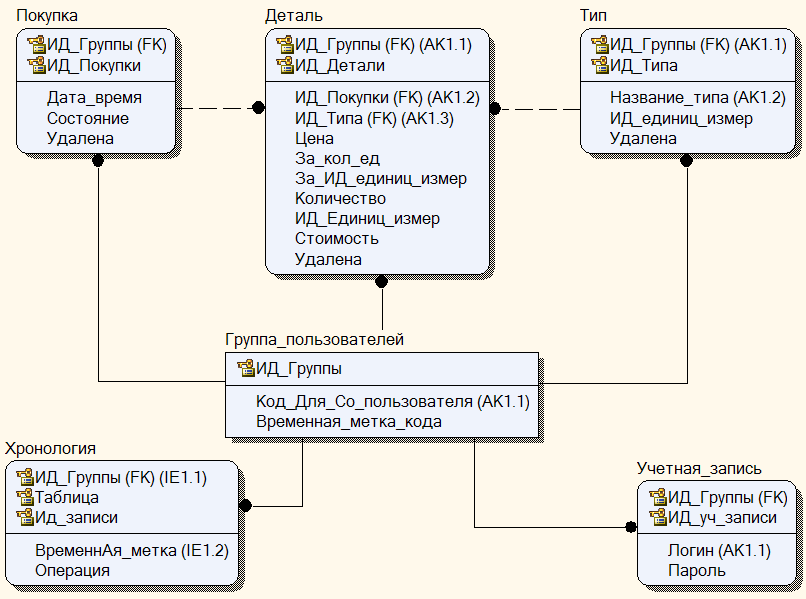


Рисунок 3.3‑2. Диаграмма БД сущность связь подсистемы сервера.

СУБД MySQL поддерживает не уникальные индексы (IE - Inversion Entry (non unique AK)), поэтому для ускорения поиска записей был добавлен индекс в таблицу хронология. Остальные индексы были добавлены для уникальности записей.

## Требования к производительности

## Ограничения проектирования

## Атрибуты программной системы

## Дополнительные требования

# Сопровождающая информация