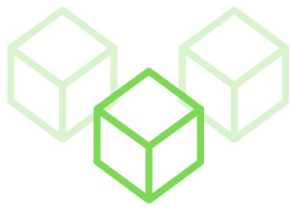


# metassist



Enjoy life in the modern world  
let us do the rest



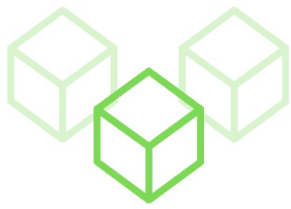
# metassist

## 1. О проекте

Если в двух словах – цель проекта, – создать приложение, которое поможет пользователю спланировать свой день до мелочей: от транспорта на работу, до чистки зубов с утра. Конечно, только от человека зависит детализированность планирования: принять ли чистку зубов за одно событие, или разделить на отдельные события чистку верхних и нижних зубов

На сегодняшний день существует огромное множество аналогичных приложений: от обычных записников типа TODO, до нагромождённых календарей, с возможностью синхронизации с другими сервисами, однако сложно сказать, что есть такое приложение, которое бы действительно завоевало популярность, и стало бы удобным если не каждому, то большинству. Возможно, это потому что, как правильно, такие приложения плохо «адаптированы» с психологией человека, и в каком то смысле, остаются оторваны от реальности. В таких приложениях, как правило, пользователю всё равно приходится всё время держать в голове много информации, чтобы оптимально им пользоваться. Например, если пользователь, – это студент, у которого по средам 3 пары. Он занесёт их в подобное приложение. Кроме того, он скорее всего выделит в календаре время для транспорта с дома до университета. Однако, если вдруг, случится так, что первую пару отменяют, пользователю не достаточно будет просто отменить событие «1 пара» в планировщике, ему нужно будет ещё подумать о:

- отмена события «транспорт», так как вряд ли человек поедет в университет на ранее намеченное время
- добавить новое событие «транспорт», на 2 часа позже
- прикинуть чем бы себя занять на те 2 часа, которые он неожиданно проведёт дома (звучит просто, но на деле человек не всегда способен быстро придумать себе чем бы заняться в ближайшее время, даже если есть много дел)
- подумать о еде: если у пользователя имеется событие «обед» где-то между, или после пар в университете, оно скорее всего должно быть



# metassist

подвинуто, так как завтрак у него будет позже в этот день, чем планировалось

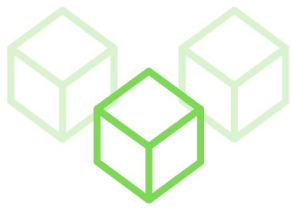
- и тд и тп...

А что, если бы всё это делалось автоматически, достаточно только отменить первую пару в планировщике, и день сам перестроится?

Ещё пример: возможно, человек ведёт здоровый образ жизни, и следит за своим питанием. Но к тому же, он занятой человек, и следит за своим временем. Есть много специализированных приложений, позволяющих на высоком уровне следить за своим питанием: как часто есть, какие продукты есть, где есть и тд. Допустим, у пользователя есть такое. Кроме этого, он пользуется каким-то планировщиком своего времени, о которых мы рассказывали выше. Но приложения не будут знать друг о друге. И пользователю самому придется держать в голове то, что ему предлагает приложение с едой, и что у него записано в календаре, так как первому приложению всё равно на то, что в планировщике у человека с 12:00 до 13:00 встреча, если пришло время кушать, оно скажет ему об этом, и тогда пользователю придется самостоятельно переносить в нём обед до лучших времён. Это всё неудобно, особенно, если учесть, что еда – далеко не единственное занятие у людей на протяжении дня. Та и кроме того, допустим, если ужин у человека в этот день будет состоять из яичницы, то человеку нужно помнить, что ему нужны будут яйца, и пока он не пришел домой, было-бы неплохо зайти в магазин, и купить их. Конечно, быть может, приложение с едой наметит человеку, что ему нужны будут яйца, но оно точно не внесёт в его календарь событие «зайти в магазин», не рассчитает время на транспорт до магазина, не перестроит план на оставшуюся часть дня, в связи с тем, что пользователю ещё придется заходить в магазин, и он придет домой позже.

И последний пример:

Скорее всего, всем знакома ситуация – начали одно большое дело, например читать книгу по C++ на миллион страниц, не дочитав и до конца первого раздела, попросту забываем о ней, переключаясь на какие-то новые задачи. А через пару недель вспоминаем о книге, и понимаем что уже не помним что там было в ней в начале, и приходится или перечитывать сначала, или откладывать на будущее. В результате – куча потерянного времени. Конечно, выход очевиден: просто пользоваться каким-то планировщиком, и там постоянно вносить событие «читать книгу C++» длинной в пару часов, но если таких «больших дел» у вас много, то в разные моменты времени вы будете по разному смотреть на разные задачи, и, сами того не замечая, будете выделять всё меньшие и меньшие интервалы времени на C++. Конечно, вы можете задать в планировщике регулярную задачу «По средам с 20:00 до 22:00 читать книгу по C++». Однако, с появлением походу чтения всё новых и новых задач, вам всё больше будет казаться выделение такого большого интервала времени на C++

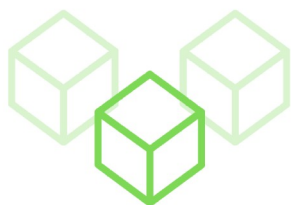


# metassist

расточительством, А всё потому, что вам приходится самостоятельно держать в голове все те большие задачи, которые перед вами сейчас стоят: C++, видео-курс по Rust, дипломный проект. И в разные дни вам больше будет хотеться заниматься определёнными действиями, так как будет казаться что вы его забросили больше остальных, всё это вызывает панику, и нежелательное чувство «что вы ничего не можете закончить до конца, и вряд-ли когда-то закончите» Далее мы расскажем, как metassist призван решить эти и другие проблемы, и стать действительно незаменимым ассистентом человека на постоянной основе.

В metassist наша команда пошла по другому пути: мы не предлагаем людям продукт – «абстрактный планировщик их дел», мы предлагаем решение их реальных и ежедневных проблем. Для этого, попытаемся сформулировать эти проблемы, которые не удастся решить обычным планировщикам:

1. Здоровый и постоянный сон. Как правило, люди которые хоть как-то планируют своё время, просто ограначиваются жесткими указаниями самому себе дескать «ложиться ровно в 23:00, вставать в 7:00». Более усердные в этом плане, могут внести событие «сон» в планировщик, и, скорее всего, добиться того-же результата, что и люди первой категории, а конкретнее – никакого. Всё равно большинство ложится в пол второго, и просыпает каждое утро. И всё потому, что сон в том же планировщике никак не коррелируется с делами людей: если человека уже достало, что он постоянно откладывает книгу по C++, он может сказать себе: «ну всё, сегодня не буду спать до утра, но прочитаю минимум половину». Конечно, каждому человеку самостоятельно решать, что для него правильно, и вполне возможно, что читать до утра для многих – это прекрасно. Однако, если они запланировали сон с 23:00, когда их мозг мыслит более трезво, и небыл разбит собственной неспособностью осилить книгу, то чтение вместо сна определённо будет плохой идеей для такого человека
2. Занятие спортом. Большинство в наше время так или иначе занимаются спортом. У всех разные подходы: бег с утра, вечерняя тренировка в зале, бассейн после пар, и тд. Но в любом случае, идеальное планирование тренировок – это очень сложный процесс, которым как правило довольствуются или профессиональные спортсмены, или как минимум люди, занимающиеся спортом уже не первый десяток лет. А всё потому, что нужно учесть массу факторов, например:
  - обязательно нужно плотно поесть за полтора-два часа до тренировки
  - обязательно поесть **сразу** после тренировки

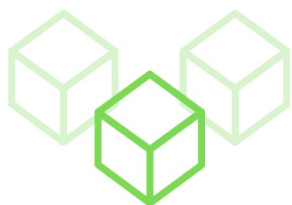


# metassist

- регулярность занятий: например вт, чт, сб (если занятие отменяется по какой-то причине, возможно, потребуется перестроить занятия на этой неделе, например провести их в ср, пт)
- Начало тренировки не позже 18:00
- завязанность тренировок на каких-то других событиях: например, в случае отмены тренировки, не назначать её перед работой, а переносить только на вечернее время

Для занятого, работающего человека сложно держать все эти факторы в голове, и планировать всё оптимально. Часто человек едет на тренировку сразу после работы, с мыслью «если сейчас зайду домой поесть, то точно уже не поеду на тренировку, лучше идти сейчас», при этом последний прием пищи был 6-7 часов назад. В подобных условиях, такие тренировки принесут наверное 5% результата, который могли бы принести. А ведь можно было поесть рядом с работой, за час до выезда оттуда, и приехать на тренировку как раз полным сил

3. Прием пищи. Ещё одна довольно частая проблема у современных людей. Пожалуй чаще всего, сейчас встречается такой сценарий: просыпаясь с утра, человек в спешке успевает съесть какой-то бутерброд, или, если повезёт, яичницу. Готовить себе еду на работу/в институт времени не остается. Или продуктов нету, или просто уже время выходить, или с утра плохо соображает голова что готовить. На работе иногда человек идёт в какое-то кафе, там, как правило, питается не лучшей едой, тратя деньги, за которые мог бы приготовить себе еду получше, а иногда просто терпит до вечера, приезжая домой, и там уже что-то готовя. А вечером ещё и оказывается, что нету продуктов приготовить то, что хочется, а желания идти в магазин нету и подавно. Остается надеяться, что человек не закажет доставку еды из того-же кафе, где ел днём.
4. Уборка дома, и другие обязанности по дому. Если проанализировать, почему это часто является такой проблемой для людей, становится понятно, что причина скорее не в том, что это забирает много сил, а в том, что работа появляется неожиданно, и в не самое подходящее время, в следствии чего, людям начинает казаться, что уборка длится бесконечно, и появляется желание не делать её любыми способами, ведь «сейчас уберу, а потом опять что-то придется делать». Быть может человек хотел пойти в кино, а тут вдруг оказывается, что нужно убрать на балконе. Начинается или откладывание уборки, с чувством вины, или отмена кина. В любом случае настроение испорчено. Выход очевиден – сесть, и продумать всю работу, которую нужно делать в доме, и сделать её

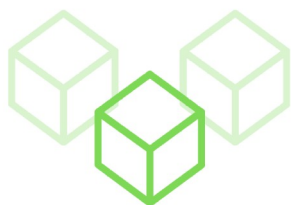


# metassist

регулярной, назначая на определённые дни недели или месяца. Тогда уборка станет намного менее болезненным событием, так как будет приходить запланированно, и все члены семьи будут ожидать этого, и будут знать что это нужно будет делать в такое. И ничего не будут назначать на время работы по дому. Чувство бесконечности работы исчезнет

5. Транспорт. Это не столь острая проблема, как вышеперечисленные, однако тут тоже есть о чём подумать. Ситуации, когда мы не совсем оптимально передвигаемся по городу случаются часто. Например, с утра я поехал на работу, днём приехал домой, вечером решил поехать в магазин в районе работы. Конечно, было-бы лучше, заехав в магазин после работы, а потом уже ехать домой. Ещё одна проблема, связанная с транспортом, заключается в том, что транспорт – отличный способ занять себя чем-то. Есть целый ряд вещей, которые удобно делать, пока вы едете домой. Вместо того, чтобы просто слушать музыку, можно, к примеру, перебирать карточки с иностранными словами, и учить новый язык. Но для такого нужно планирование: чтобы это было на регулярной основе, и с привязкой к времени езды. Например, если вы едете домой 15 минут, из которых 10 вы сидите в маршрутке, было-бы неплохо запланировать такое занятие, которое требует мало времени (10 минут). В ручную это, как правило, никто не делает. Ведь каждое утро нужно держать в голове сколько времени на транспорт вы потратите, и соответственно планировать себе занятие. Как результат, тратится куча времени в пустую, или используется не лучшим образом
6. Неспособность в освобождающиеся небольшие промежутки времени занять себя нужным делом. Типичный пример: перемены между парами, как правило тратятся в пустую. А это, в сумме, приблизительно, час времени каждый день! Человеку сложно сориентироваться чем бы ему заняться прямо сейчас, кажется что времени слишком мало, и большинство людей ограничиваются пустой болтовней под кабинетом в ожидании следующей пары. Опять таки: сложно постоянно держать в голове все свои задачи, и назначать их себе, особенно, если время появляется внезапно
7. Сложности с правильным планированием времени на большие задачи (проблема из 3 примера обычных планировщиков). Когда у человека с десятком огромных заданий, начиная от изучения японского языка, заканчивая той же книгой по C++, которые он сам перед собой поставил, становится сложно «администрировать» их выполнение. Проблема





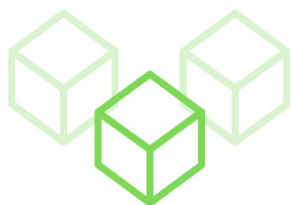
# metassist

заключается в том, что пользователю приходится **постоянно** самостоятельно назначать выполнение какой-то большой задачи. А в разное время у человека может быть разное представление про необходимые траты времени на каждые задачи, потому не будет никакой системы в выполнении каждой из них

Мы перечислили несколько основных проблем современного человека. Как уже говорилось раньше, metassist – это не планировщик времени, это средство, позволяющее решать вышеперечисленные проблемы, и многие другие. Как можно видеть, это намного больше, чем просто планировщик.

Теперь, более детально опишем приложение и его возможности, а так же способы, которыми будут решаться вышеперечисленные проблемы

Metassist — это графическое приложение, которым можно будет пользоваться с любого устройства даже без доступа к интернету. Основной концепт приложения, отличающий его от аналогов: пользователь **не** вносит конкретные задачи в ежедневник на конкретное время. Как правило, он вносит обобщённые задачи, которые требуют некоторое время на выполнение, и ряд ограничений: разрешенные места проведения, разрешенное время выполнения, условия на основе результата выполнения других событий, и множество других ограничений, которые может использовать пользователь по желанию. А metassist уже сам выбирает самый оптимальный план выполнения этих задач, показывая пользователю план на текущий день. Вот и получается, человеку не нужно ни о чём думать, система сама помнит про все события, а так же причины их назначения, и в случае отмены, примет решение, подходящее для конкретного события. И никакие длинные события, выполняемые много месяцев не забудутся



# metassist

## 2. Описание возможностей

Общий принцип работы системы следующий: пользователь одним из множества возможных способов вносит некоторую информацию про свои занятия, и задачи, которые необходимо выполнить. После этого, он в любой момент может просматривать план своего дня в виде календаря. То есть, основные возможности, которые доступны пользователю:

- внесение информации про свои задачи
- просмотр плана, построенного на основе внесенной ранее информации

Предположим, пользователь только скачал metassist, зарегистрировался, и зашел в приложение первый раз. Он увидит пустой план на сегодня, так как задач ещё нету, и несколько служебных кнопок. Одна из них – добавить новую задачу. По нажатию на неё, пользователь сразу должен будет выбрать в открывшемся меню тип задачи. Всего будет доступно 4 типа задач:

- одиночная
- регулярная
- сумарная
- блок

После выбора типа задачи, откроется окно для внесения информации про новую задачу, которое состоит из нескольких полей, в зависимости от типа задачи. После окончания, приложение начинает генерировать новое расписание, встраивая в него появившуюся задачу

Расскажем в общих чертах про каждый тип задач

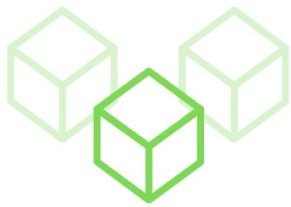
Одиночная задача – самый простой тип задач. Это задачи, которые случаются единожды, и имеют четко обусловленное время начала и время конца.

Например: встреча с работодателем, поезд на природу, поход в магазина

Регулярная задача – как понятно из названия, это задача, которая происходит регулярно, с каким-то интервалом. Например душ, сон, пары в университете, работа, тренировки. При добавлении новой регулярной задачи, пользователю будет доступно 2 возможности:

- Задать конкретное время начала и время конца события с некоторой регулярностью. Например: «каждый вторник и четверг с 15:00 до 17:00, а так же каждую субботу по нечётным неделям с 12:00 до 15:00 – плавание



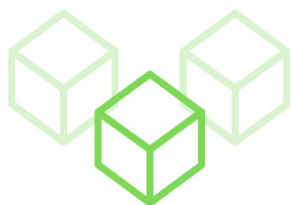


# metassist

- в бассейне», ещё пример: каждое 12 и 15 число месяца с 11:00 до 16:00 встреча с детьми
- Или же задать интервалы, в которые могут происходить события, и размер во времени этих событий. Например: каждую неделю в субботу или воскресенье с 10:00 до 20:00 нужно попылесосить, что занимает полтора часа. Это вся информация, которую должен задать пользователь. И metassist сам встроит это событие на самое подходящее время, в соответствии с остальными задачами. И будет делать это каждую неделю, быть может в разное время. Таким образом, пользователь ни о чём не забудет. Ещё пример: «Я хочу учить немецкий язык. Вот когда я хочу это делать: в понедельник с 10:00 до 15:00 – один час, во вторник с 10:00 до 13:00 или с 15:00 до 20:00 – два часа, и каждую третью неделю в среду с 10:00 до 22:00 – три часа. Причём событие можно разрывать, но так, чтобы один подход был не короче одного часа, но и не дольше двух часов». При задании такой информации, metassist подберёт на каждую неделю самое удобное время для немецкого языка. При необходимости, событие будет разорвано, но только в рамках интервала. В этом примере у нас 3 интервала: каждый день — это новый интервал. Таким образом, например, во вторник пользователь, возможно, с утра до 13:00 будет учить 1 час немецкий, а вечером ещё один час. Или быть может оба часа с утра. Но, гарантируется, что за вторник он в общей сложности будет делать это 2 часа. Именно возможность задания таких «интервальных» событий является одной из отличительных особенностей metassist, так как это значительно приближено к реальной жизни человека: возможно он не знает когда конкретно на каждой неделе во вторник у него выйдет учить немецкий язык, но он хочет быть уверен, что metassist помнит о том, что он хочет делать это именно 2 часа.

Конечно, человек может смешивать обе возможности. И для одного действия задавать как и чёткое время выполнения, так и дополнительно интервалы, в которые событие так-же должно происходить

Сумарная задача – самый интересный тип задач. Внешне форма для добавления сумарной задачи полностью идентична форме для регулярной задачи. Но дополнительно там есть 2 поля: необходимое время и дедлайн. Механизм сумарных задач работает по следующему принципу: вы вносите в metassist несколько больших задач, указываете сколько вам потребуется времени в часах на их выполнение, и до когда вы хотите в этом преуспеть. И как только, у вас появляется свободное от одиночных и регулярных задач время, приложение выбирает сумарную задачу, которую лучше всего сейчас выполнять, и говорит вам об этом(). Сумарные задачи – это все те большие начинания, которые мы бросаем даже не дойдя до середины. Кроме того, это все события, которые



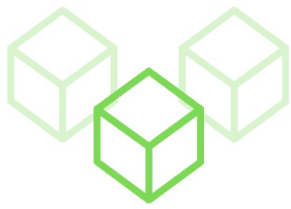
# metassist

требуют нашего многократного вмешательства, но для которых мы не хотим создавать чёткий график выполнения. При помощи metassist и сумарных задач можно указать несколько таких начинаний и событий.

Например: допустим сегодня 1 октября 2019 года. Вы добавляете 3 новых сумарных задачи, со следующим общим временем и дедлайном:

Задача:	Нужно времени:	Дедлайн:
Научиться играть на гитаре	500 часов	31.01.2020
Написать тесты для приложения	20 часов	20.10.2019
Пройти курс по node.js	70 часов	10.11.2019
Прочитать книгу по C++	150 часов	31.12.2019

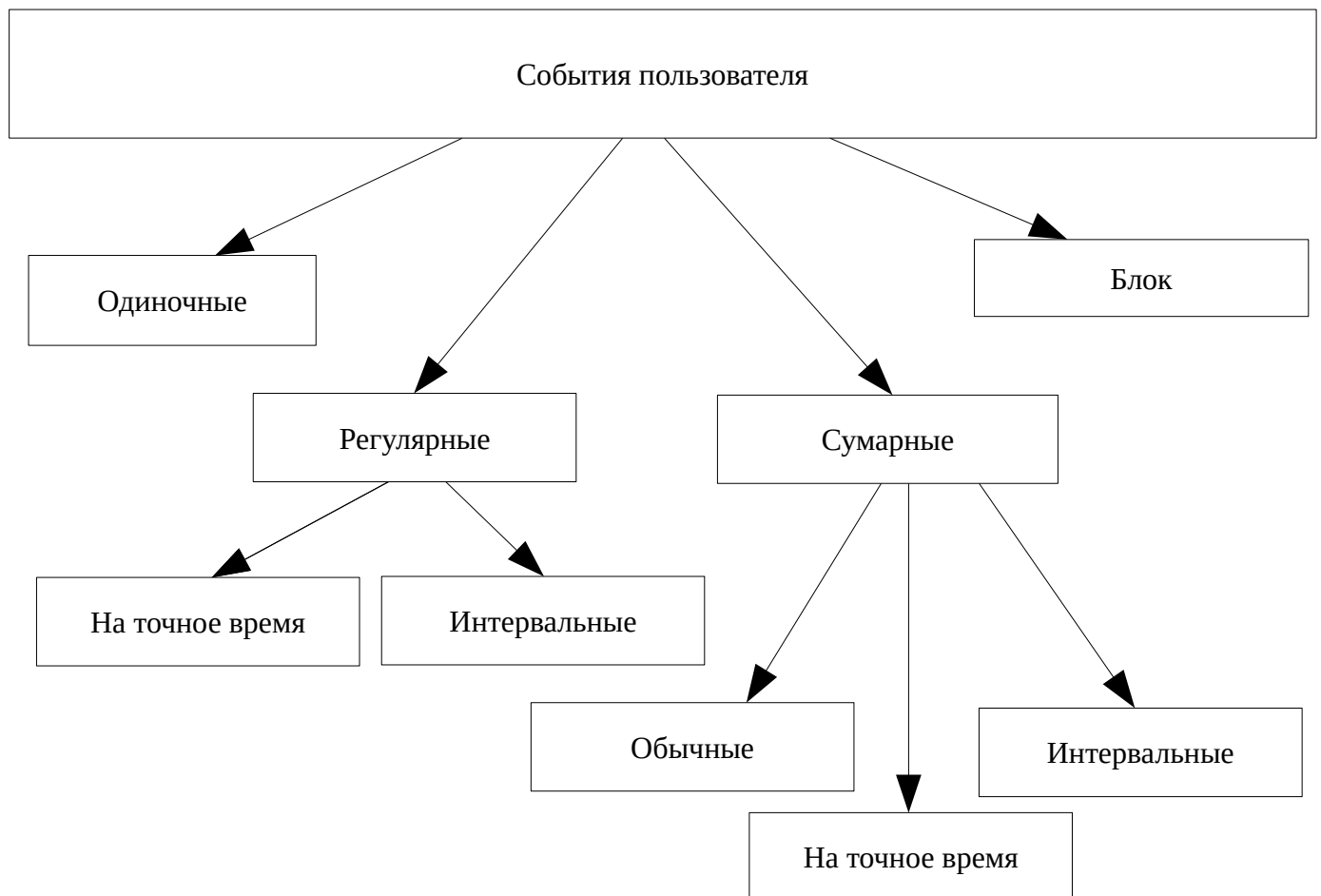
Теперь metassist никогда не забудет ни об одной из этих задач, пока вы не дойдете по каждой из них до нужного времени выполнения. В методе построения оптимального календаря есть специальный алгоритм, который выбирает самую срочную сумарную задачу в соответствии с её ограничениями на каждый промежуток свободного времени. По истечению необходимого времени на выполнение, metassist предложит вам: или считать задачу выполненной, или добавить ещё некоторое время на её выполнение, если вы чувствуете, что пока не справились с задачей, и, если нужно, отодвинуть дедлайн. Что важно, не может случиться такой ситуации, при которой дедлайн уже наступил, а вы не успели потратить указанное вами время на выполнение этой задачи. Потому-что ещё на этапе добавления задачи metassist вам скажет, что вы добавляете задачу с невыполнимыми требованиями, которую вы скорее всего не сможете успеть выполнить в срок. А если же вы будете слишком часто отменять запланированные для вас «подходы»(то есть один конкретный раз, когда вы выполняете сумарную задачу) к выполнению задачи, часы со счётчика часов задачи сниматься не будут, и в один момент metassist уведомит вас о том, что вы слишком медленно выполняете задачу, и такими темпами рискуете не успеть до дедлайна. Кроме того, он всё чаще начнет вам в свободные промежутки ставить именно ту задачу, которую вы отменяете, так что вам всё равно придется её выполнять. Кроме того, в форме задания сумарной задачи есть все те же поля, что и у регулярной. То есть, вы можете для сумарной задачи указать в какие часы вы хотите выполнять её, или, например, по каким дням недели сколько часов, в общем все те же возможности, которые есть в регулярной задаче, есть и у сумарной

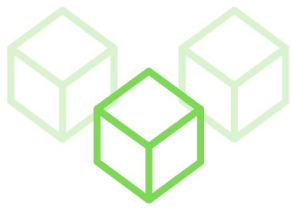


# metassist

И последний тип задач – это «блок». Это специальный тип задач, который часто нужен обычному человеку для автоматизации его рутинных задач. Блок — это просто набор идущих подряд регулярных задач, как правило маленьких, которые можно добавить один раз в блок, и уже оперировать с блоком, а не с каждой из маленьких задач. Например, каждое утро я встаю в 6:00, и постоянно делаю одни и те же действия: зарядка, чистка зубов, завтрак. Потому я могу всё это скомпоновать в один блок. И в будний день уже добавить этот блок, а не каждую задачу по отдельности. При создании блока вы можете указать время его начала, а затем просто указать сколько минут нужно на каждую подзадачу в нём. Или же не указывать время начала. Тогда при размещении в конкретный день, вам нужно будет выбирать на какое время ставить блок в этот раз

Итак, мы перечислили все типы **пользовательских** событий:



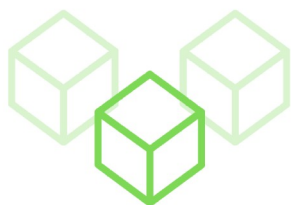


# metassist

Как уже упоминалось выше, любую задачу можно отменить, в случае, если человек по каким-то причинам не успевает её сделать. Это вызовет перестроение календаря дальше, и, возможно, отмену некоторых других задач. Кроме того, у пользователя будет возможность сообщить программе, что некоторое время, начиная с текущего момента, человек будет чем-то занят, причем он не хочет заносить информацию о том, чем именно он будет занят. Скорее всего из-за того, что это долго. Например, человеку внезапно понадобилось встретиться с кем-то. Тогда, чтобы metassist знал о том, что пользователь ближайшее время не будет выполнять задачи из планировщика, человек просто зайдет в приложение, нажмет одну кнопку, и укажет сколько времени он будет занят, например 2 часа(при этом, по желанию, можно указать чем он будет занят, чтобы потом можно было смотреть статистику занятости). И metassist просто отменит все задачи на ближайшие 2 часа, и при надобности перенесёт их. Очень удобно и быстро.

У пользователя так же будет возможность задавать события, выполнение которых может начаться только после выполнения другого события. Это будет работать не со всеми типами задач, а только с теми, для которых это действительно может понадобиться. Несколько примеров:

- Допустим, человек изучает Erlang посредством прохождения онлайн-курсов. Всего есть 3 курса: Erlang starter, Erlang essential, Erlang advanced. Конечно, прохождение каждого из них может начаться только после завершения предыдущего. Всё это сумарные задачи. Если мы их захотим внести в metassist одновременно, то мы не будем знать дату начала прохождения второго и тем более третьего курса, так как не известно когда мы закончим предыдущие. Вариант указывать датой начала следующего курса дедлайн предыдущего – плохой способ, так как вполне естественно, что пользователь закончит прохождение предыдущего курса раньше дедлайна, и тогда ему не захочется ждать дедлайна, только после которого начнется следующий курс. Потому, для удобства, он сможет указать временем начала прохождения каждого курса время окончания предыдущего. Причём таким образом можно выстраивать целые цепочки из желаемых задач, зависящих от других задач. Единственное, за чем пользователю придется следить (metassist подскажет об этом), – это дедлайны следующих за предыдущими задач. Так как если предыдущая задача была закончена значительно раньше, или значительно позже её дедлайна, это скорее всего отразится на дедлайне следующей за ней задачи. В любом случае, при начале новой сумарной задачи, metassist попросит человека уточнить новое время дедлайна, или оставить его таким как есть



# metassist

- Заехать в магазин после работы. В данном случае, работа – это регулярная задача, а магазин – одиночания. Однако, если человек чувствует, что в этот день его могут задержать на работе на неопределённый срок, он не сможет просто указать время поездки в магазин то время, когда у него заканчивается рабочий день. Потому временем поездки в магазин указывается не определённое время типа 18:00, а время окончания работы в этот раз

Ещё одна полезная опция, которая будет доступна человеку – создавать запланированные диалоги, касательно каких-то событий. Это означает, что при добавлении новой задачи вы знаете про неё слишком мало. И просто хотите её внести в каленарь, чтобы не забыть про неё. Но вы знаете, когда у вас появится больше информации про неё. И на это время вы можете запланировать уведомление metassist в виде диалога, который напомним вам о том, что вам нужно внести дополнительную инфомацию про неё. Проще всего показать полезность этой возможности на примере, для которого, в основном она и предназначена – питание

Допустим, при планировании своего буднего дня (которое осуществляется один раз в виде регулярных задач, и дублируется автоматически на все или некоторые дни) вы создали приблизительно следующее:

6:40 – будильник

6:40–6:50 — сделать зарядку

**6:50–7:00 — приготовить завтрак**

**7:00–7:10 — позавтракать**

...

9:00–13:00 — работа

**13:00–13:30 — обед**

13:30–17:00 — работа

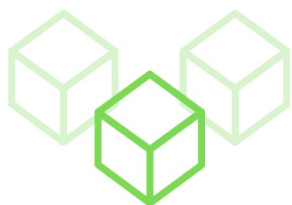
17:00–18:00 — транспорт домой

**18:00–18:30 — приготовить ужит**

**18:30–19:00 — поужинать**

...

Если не использовать запланированные диалоги, то такое планирование будет иметь большую проблему – человек каждый день есть разную пищу, которая требует разное время приготовления, а иногда не требует, если он есть еду, приготовленную раньше. Потому, человек может сообщить приложению, что каждую неделю в воскресенье в 20:00 он должен попросить его конкретизировать данные события, и изменить их. При этом, пользователю просто по очереди открываются все формы с событиями, у которых



# metassist

запланирован диалог на это время, и он может их изменять: начиная от добавления комментариев, например с информацией о том что он будет готовить, и изменением общего времени события, до отмены события вовсе, если он знает что ему не нужно будеи ничего готовить в этот раз. Таким образом, неделя ещё не началась, а у человека уже полностью распланирован каждый прием пищи на протяжении недели.

Кроме того, тут не будет лишним вспомнить про ещё одну небольшую, но полезную возможность, которая будет у человека – создание списков, например с продуктами. При описании того, что человека готовит, он может указать ингредиенты, необходимые ему для этого. А затем, metassist может вывести в удобном виде все необходимые ингридиенты на предстоящую неделю, и некоторые из них пользователь может внести в список с покупками, и затем в любой момент, когда он будет в магазине, открыть его, и купить всё по нему, таким образом, он не купит ничего лишнего, и не забудет купить что действительно нужно

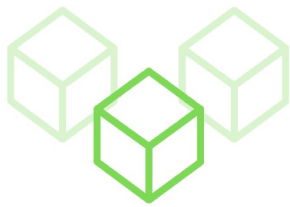
Конечно, и списки, и запланированные диалоги можно использовать абсолютно для любых целей, которые придут в голову. Питание – просто один из хороших примеров, как мы считаем

У пользователя будет и много других возможностей, сейчас мы перечислили самые основные

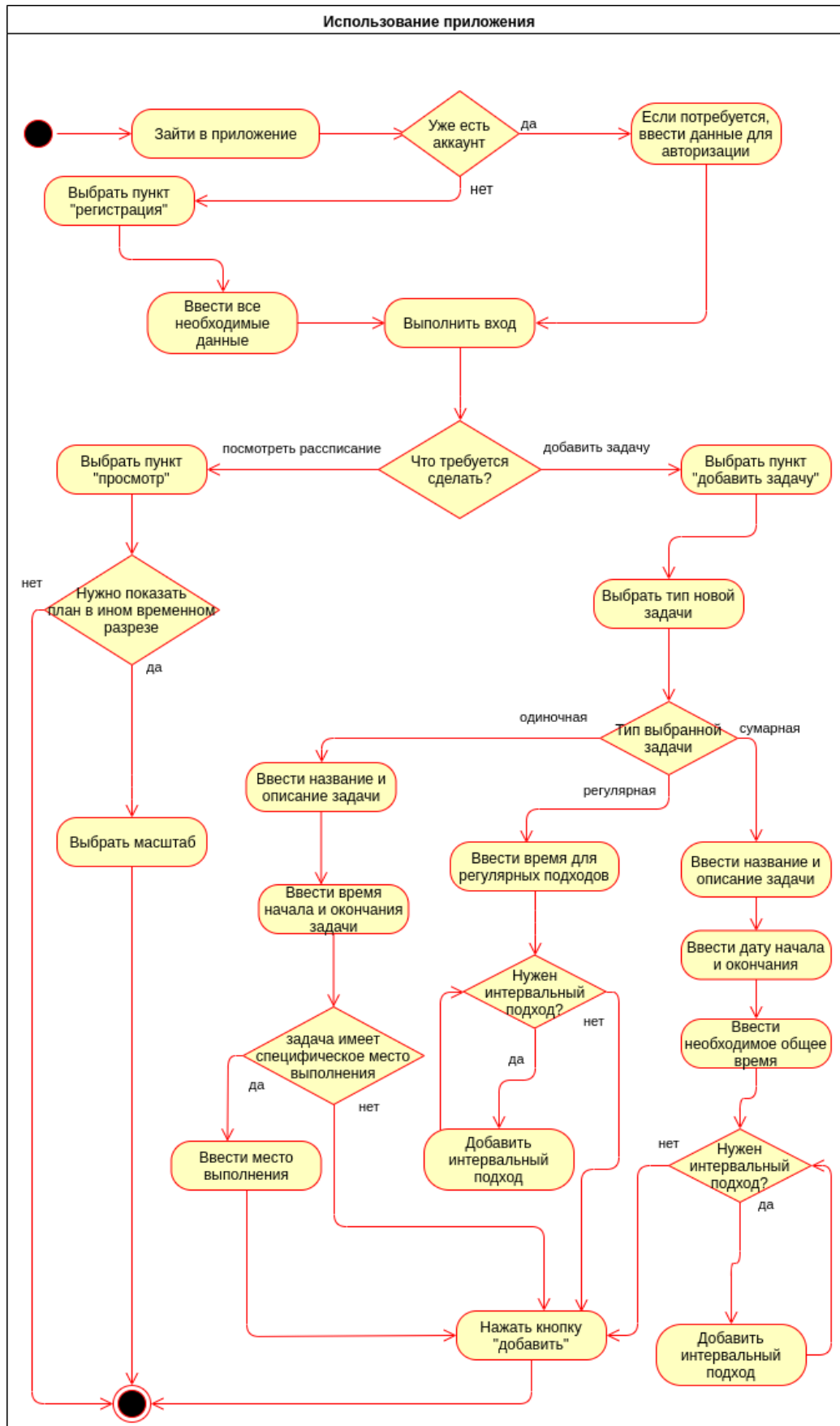
### **3. Описание БП**

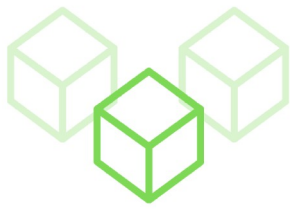
Приведём подробное описание процесса «пользование приложением», которое совершает пользователь в виде диаграммы деятельности:





# metassist

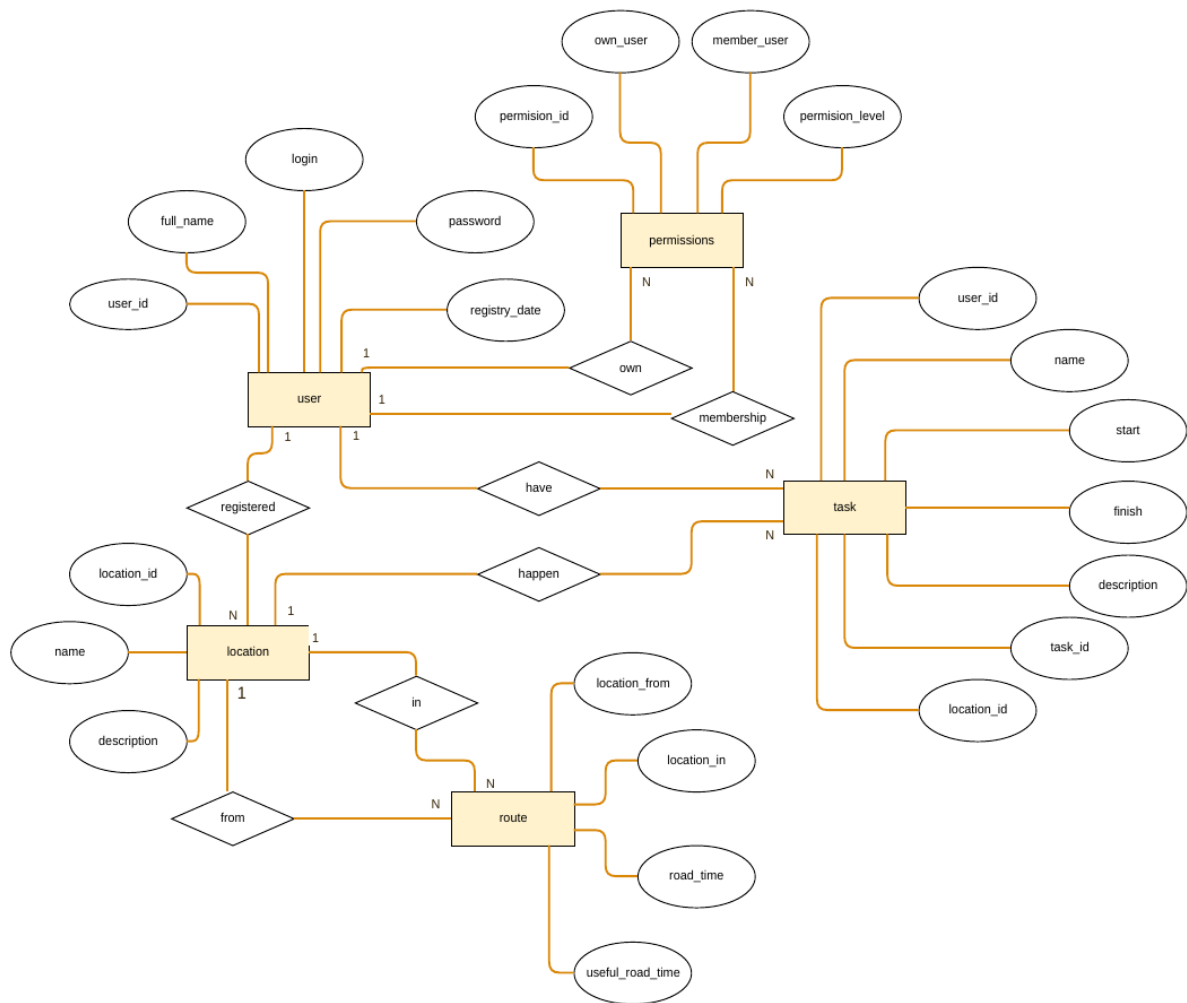


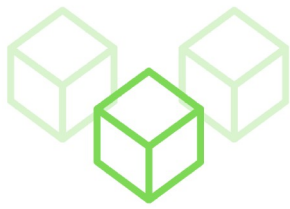


# metassist

## 4. Данные

Опишем структуру схемы базы данных в виде ER-диаграммы:

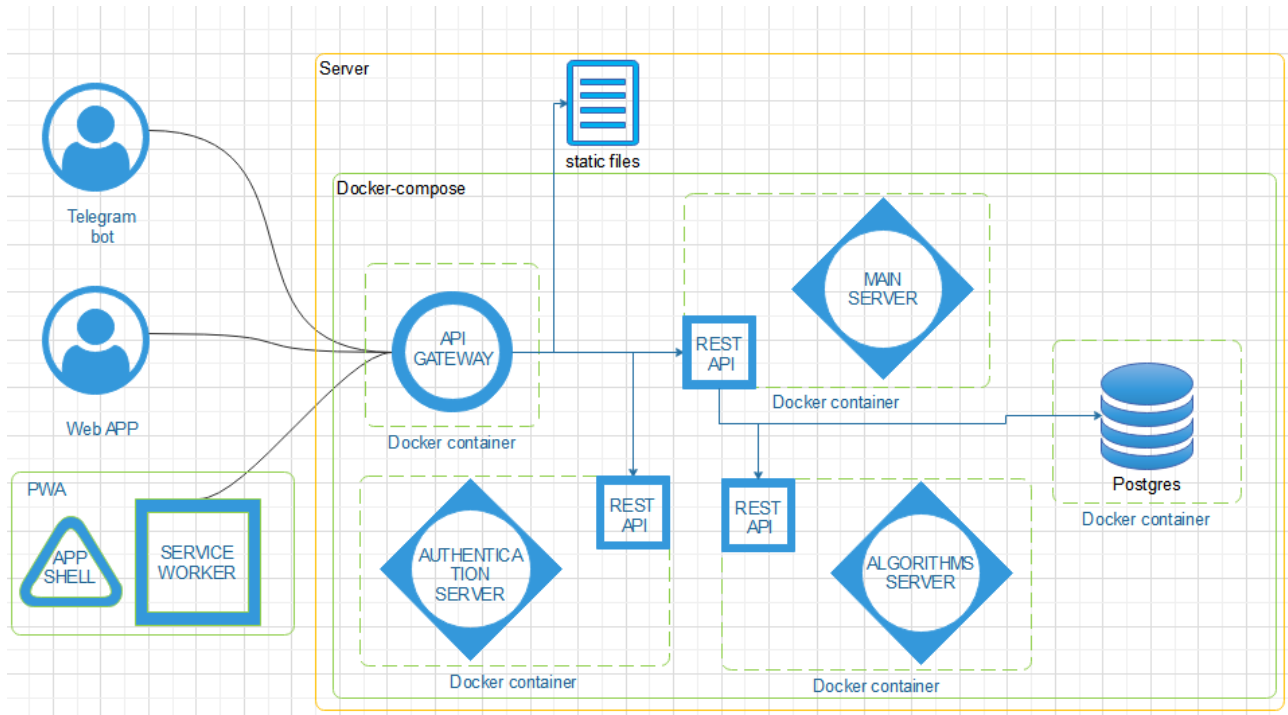




# metassist

## 5.Архитектура

Опишем архитектуру проекта в виде enterprise diagram architecture



В рамках проекта планируется реализовать телеграм-бота на Node.js, для быстрого просмотра данных о расписании дня. Браузерную версию и PWA для удобного, в том числе оффлайн, использования приложения на телефоне. Frontend будет на React. Клиенты обращаются к API GATEWAY, который распознает запрос. По-сути это сервер nginx, который отдает статический контент хранимый на сервере и по-возможности выполняет аутентификацию пользователей. При запросе динамического контента выполняет его проксирование. Основной сервер будет написан на Node.js с использованием фреймворка fastify. Будет применен паттерн микросервисной архитектуры. Каждый из сервисов, а именно сервис алгоритмов и аутентификации, будет деплоиться в отдельных докер-контейнерах и общаться с помощью REST API. Docker Compose будет использован для развертывания проекта. Динамические данные будут храниться в postgresSQL, статические - в файлах на сервере.