



Урок 2 > Сбор требований к системе

> Оглавление

- > Оглавление
- > Введение
- > Сбор требований к системе
- > Функциональные требования
 - Общее для социальных сетей
 - Основные мысли для создания социальной сети
- > Нефункциональные требования
 - САР теорема
- > Сокращатель ссылок
 - Функциональные требования
 - Нефункциональные требования
- > Хостинг текстов
 - Функциональные требования
 - Нефункциональные требования
- > Автодополнение
 - Функциональные требования
 - Нефункциональные требования
- > Облачный диск
 - Функциональные требования
 - Нефункциональные требования
- > Приложение для выкладывания фотографий
 - Функциональные требования
 - Нефункциональные требования
- > Telegram
 - Функциональные требования
 - Нефункциональные требования
- > Twitter
 - Функциональные требования

Нефункциональные требования

> Нетфликс

Функциональные требования

Нефункциональные требования

> Поиск ресторанов

Функциональные требования

Нефункциональные требования

> Сервис такси

Функциональные требования

Нефункциональные требования

> Введение

Собирать требования тяжело. Со слов заказчика вы должны понять, что ему действительно нужно.

Иногда люди не видят систему комплексно.

Помимо этого, часто ошибки начинаются с путаницы из-за *explicit* и *implicit*. Когда люди договариваются друг с другом, они могут подразумевать что-то (*implicit*), полагая, что другая сторона тоже это подразумевает. В противоположном случае можно явно перечислить всё необходимое, проговорить последовательность и способы (*explicit*).

Очевидно, необходимо заранее собирать требования к системе.

> Сбор требований к системе

Так как задача проектирования системы всегда ставится в открытой форме, нам самим нужно ограничить функционал и определить требования к системе.

Приступая к любому дизайну системы, стоит начать с ответов на основные вопросы:

- Функциональные требования — какие основные возможности дает система пользователям?

- Нефункциональные требования — насколько надежной и отзывчивой должна быть система?
- Ограничение функционала — из-за временных рамок интервью рассматриваемый функционал должен быть ограничен.

> Функциональные требования

Функциональные требования — это возможности, которые система предоставляет пользователю.

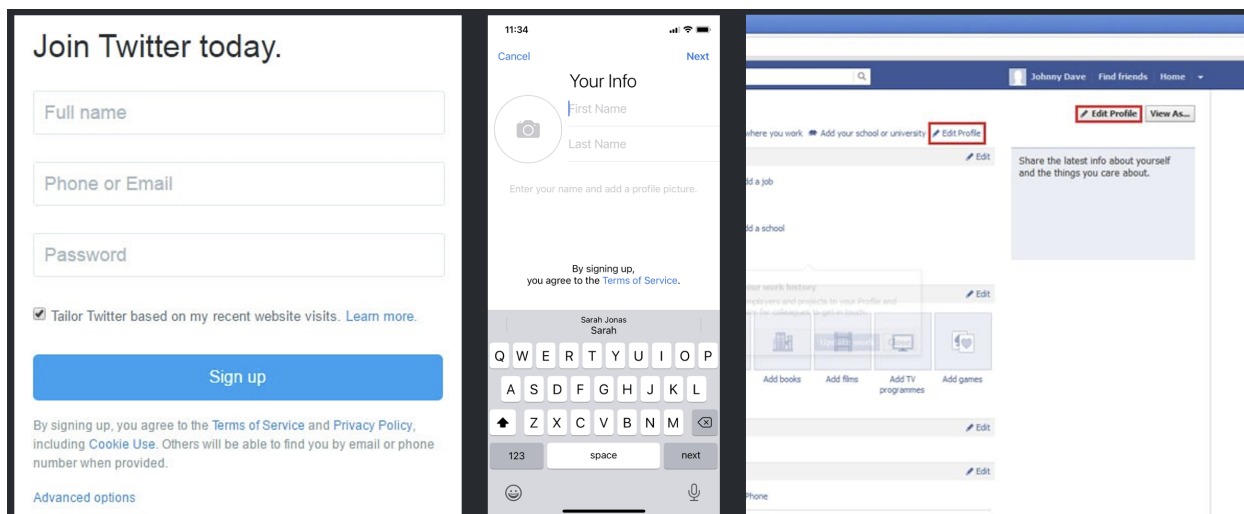
Желательно при рассказе остановиться на нескольких основных возможностях для реализации. Например, для социальных сетей это:

- добавление в друзья;
- общение между пользователями;
- создание пабликов.

Для этого стоит выписать наиболее известные свойства системы в порядке важности и провести черту под теми, над которыми мы планируем сконцентрироваться в дальнейшем. Интервьюер может помочь в данном вопросе.

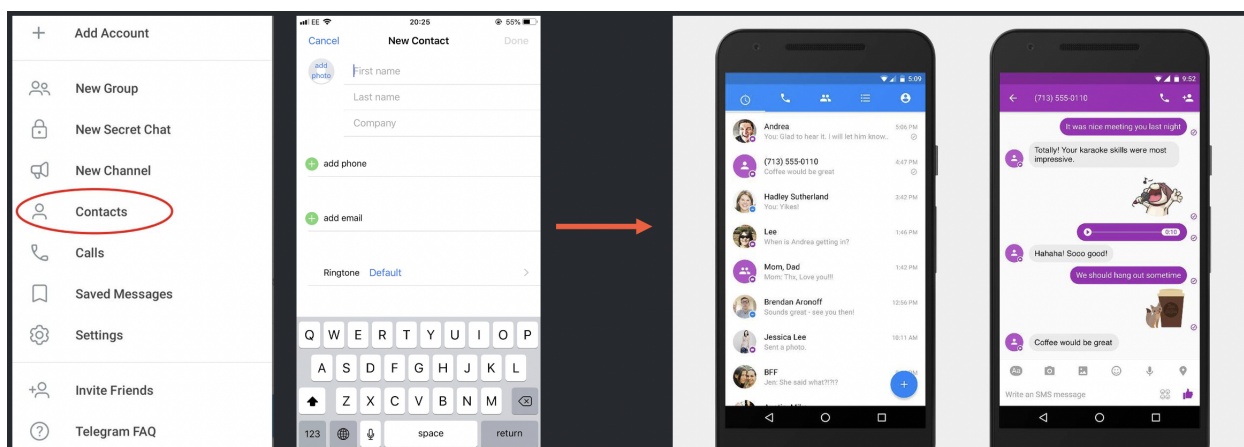
Общее для социальных сетей

- создание профиля (добавление описания и фото)

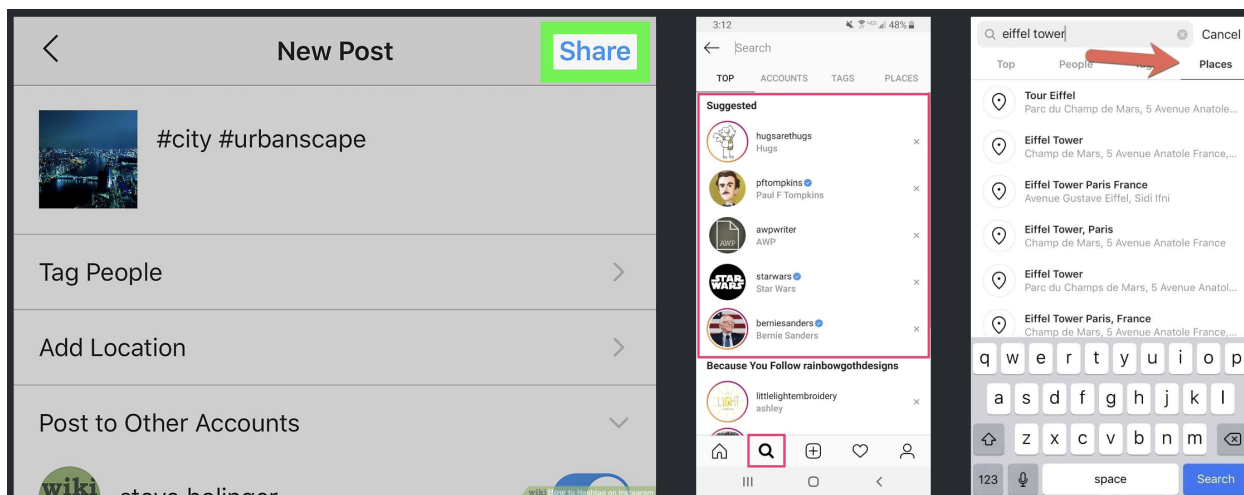


Затем, в зависимости от направленности социальной сети, можем представить такие функции, как:

- добавление контактов из сети (по номеру телефона?)
- общение со своими контактами (только текстом или с голосовыми и видео?)



Какие-то соцсети могут поддерживать все направления и дополнительные возможности, вроде добавления геотегов, поиска контента, рекомендаций и т.п.



Основные мысли для создания социальной сети

Имеет смысл выбрать тот функционал, на котором остановимся:

- создание профиля с юзернеймом;
- создание только текстовых постов;
- подписка на посты других пользователей.

А остальные возможности упомянуть, но вынести за рамки проектируемой версии:

- добавление контактов по номеру телефона;
- общение со своими контактами с видеозвонками;
- добавление тегов и геометок к постам;
- полнотекстовый поиск контента;
- отображение постов от пользователей поблизости и др.

> Нефункциональные требования

Нефункциональные требования — это то, как должна вести себя система в работе.

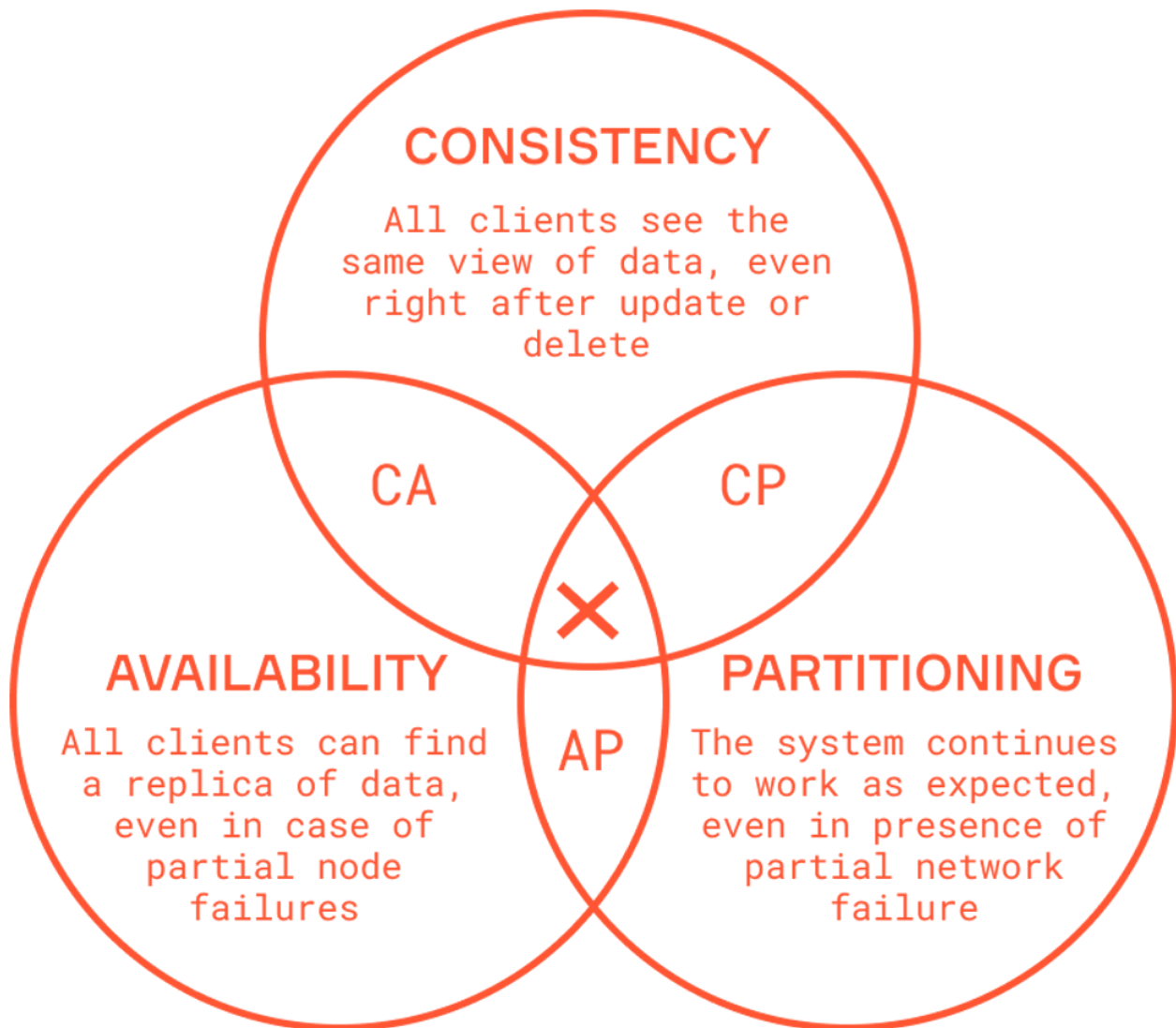
Типичные вопросы, на которые хотим ответить:

- Должна ли система быть абсолютно надежной или какие-то данные могут теряться?
- Что нам важнее — быстрый ответ системы или гарантия учета последних изменений?
- С какой задержкой могут применяться вносимые нами изменения?
- С какой задержкой отображаются запрашиваемые данные?

Changes in the Admin console or to Google services

Changing settings or performing tasks, such as managing mobile devices and users, are not always immediately reflected. Some changes can take up to 24 hours to take effect. Sometimes, you'll see a warning message if there's an expected delay.

CAP теорема



С первого взгляда все эти вопросы могут показаться странными — почему бы не сделать систему, которая удовлетворяет вообще всем требованиям? Но есть чисто теоретические ограничения, почему так не может быть, поэтому приходится всегда чем-то жертвовать.

Например, CAP теорема говорит нам о том, что нельзя создать систему, одновременно всегда доступную, консистентную и устойчивую к разделению на части (например, по сети).

В зависимости от ситуации мы делаем выбор в пользу того или иного ограничения.

Для социальной сети важно, чтобы она всегда была доступной. Социальная сеть также разделена, имея пользователей по всему миру (и сервера, соответственно).

Однако, если пользователь публикует пост, возможно, мы не сразу его увидим. Ничего страшного, если произойдет задержка, рано или поздно он все равно станет доступен для всех пользователей. То есть, консистентность не так критична.

twitter-archive/twitter-kit-android

#107 Initial loading of Tweet really slow



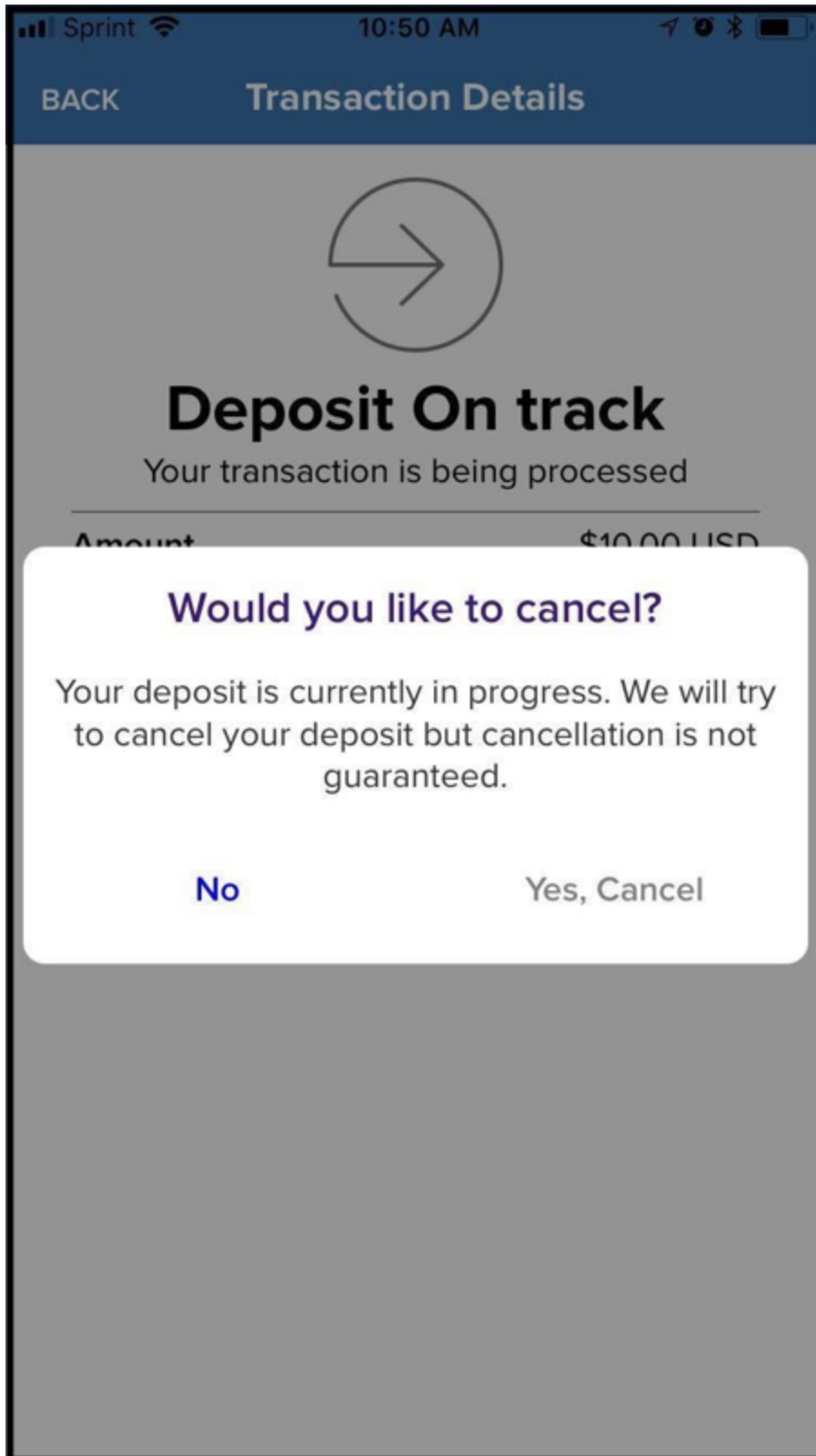
💬 2 comments



Sammekl opened on February 9, 2018



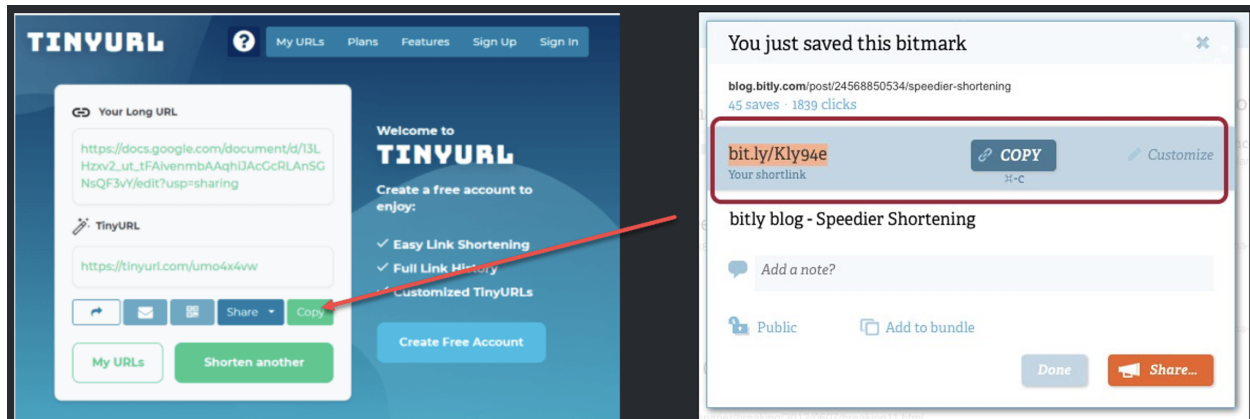
В случае с банковским приложением успешно проведенная денежная транзакция должна означать поступление и видимость средств на счету адресата, никак иначе. Ценою этому может быть недоступность приложения какое-то время.



> Сокращатель ссылок

Зачем нужен данный сервис?

Чтобы превращать длинные ссылки в короткие и делиться ими.



Функциональные требования

- По полной ссылке пользователь может сгенерировать короткую.
- Когда пользователь переходит по короткой ссылке, он перенаправляется на исходную.
- У пользователей есть возможность самим выбрать сокращенный синоним.
- Ссылки «протухают» через какое-то время. Пользователи могут его изменять.

Нефункциональные требования

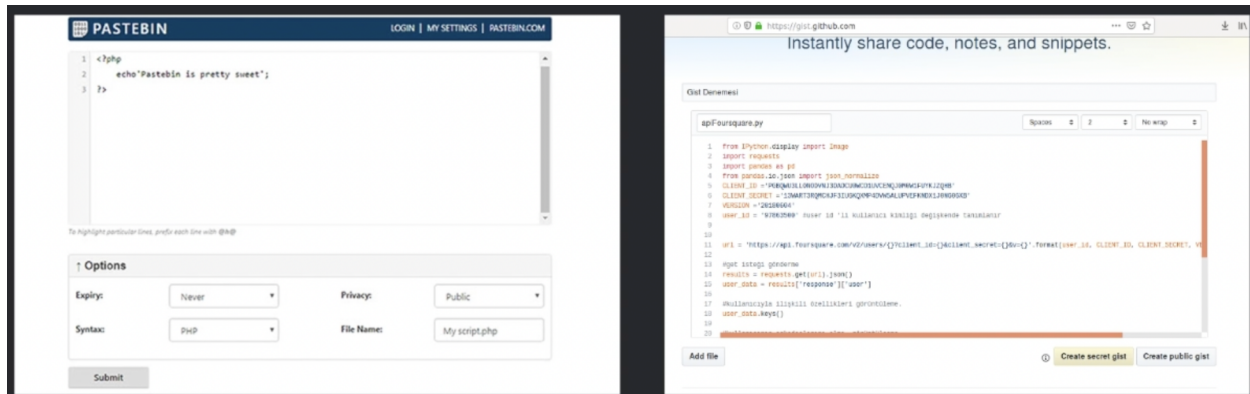
- Высокая доступность сервиса – иначе ссылки просто перестают работать.
- Перенаправление должно происходить с минимальной задержкой.
- Короткие ссылки должны быть случайны настолько, чтобы их нельзя было подобрать.

Бонус: сбор аналитики по переходам, предоставление API для разработчиков.

> Хостинг текстов

Зачем нужен такой сервис?

Чтобы делиться кусками кода и искать в нем баги вместе.



Функциональные требования

- Пользователи могут загрузить отрывок текста и получить URL для доступа к нему.
- Пользователи могут загружать только текстовые данные.
- У пользователей есть возможность выбрать синоним для доступа.
- Ссылки «протухают» через какое-то время. Пользователи могут его изменять.

Нефункциональные требования

Высокая надежность сервиса – загруженные данные не должны исчезать.

Высокая доступность сервиса – иначе доступ к нужным данным пропадает.

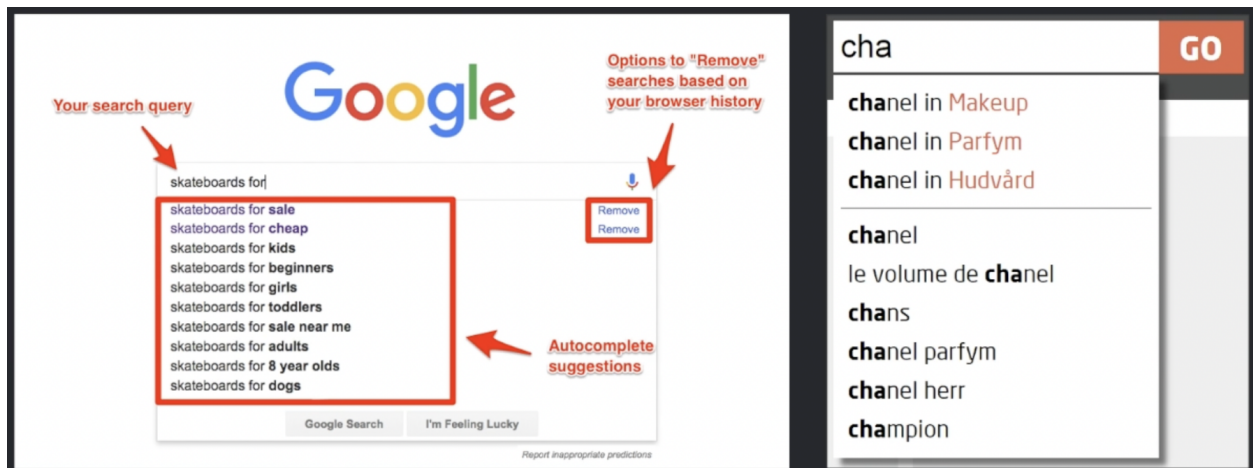
Сохраненные тексты должны отображаться с минимальной задержкой.

Короткие ссылки должны быть случайны настолько, чтобы их нельзя было подобрать.

> Автодополнение

Зачем нужен такой сервис?

Чтобы при вводе первых букв запроса предлагались варианты дополнения.
Например, при поиске «Н» сразу дополнялось до «НФТ купить».



Функциональные требования

- Пользователи вбивают запрос, для него предлагается топ-5 дополнений.
- Варианты обновляются по мере вбивания букв и слов.

Нефункциональные требования

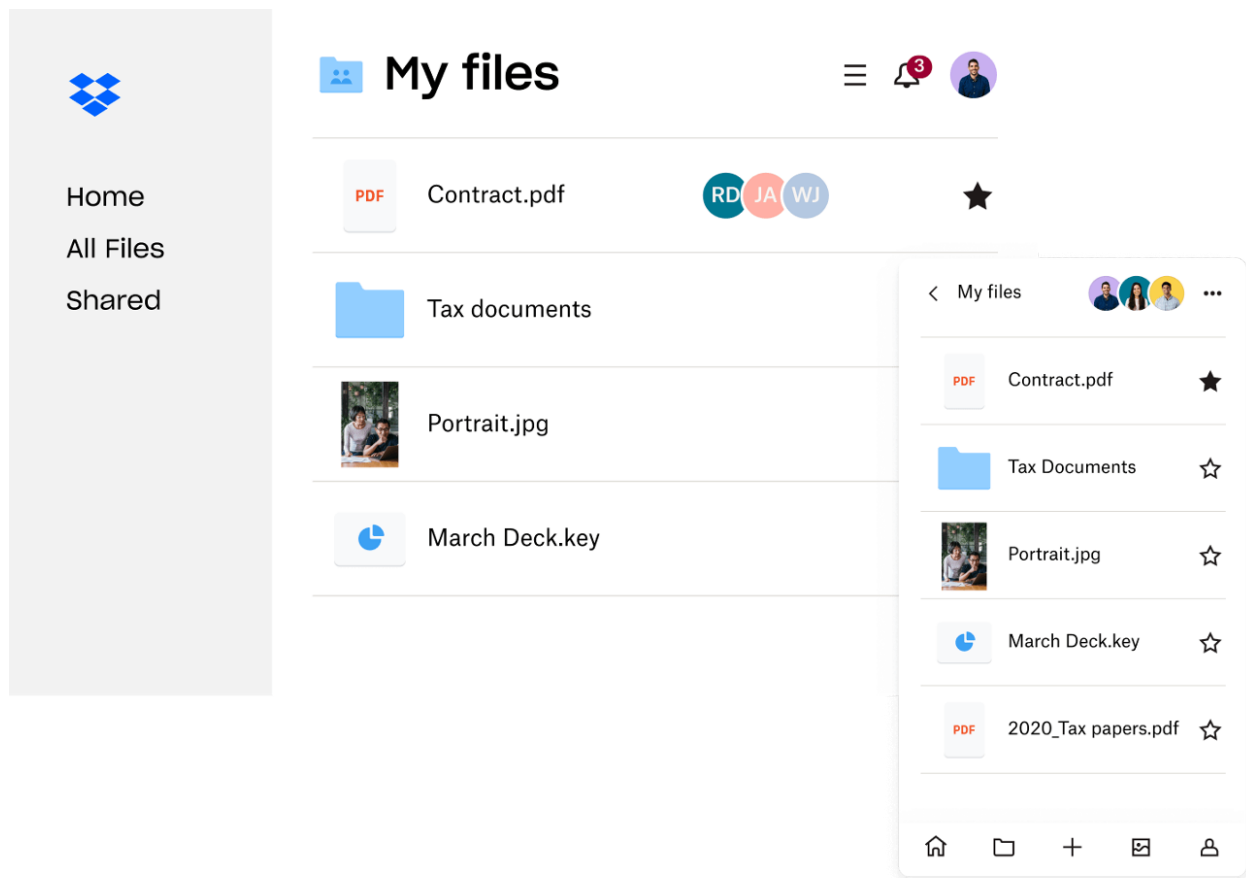
- Высокая отзывчивость сервиса – дополнения обновляются почти в реальном времени.

Бонус: учитывать прошлую историю запросов, профиль пользователя, контекст.

> Облачный диск

Зачем нужен такой сервис?

Чтобы файлы были везде доступны.



Функциональные требования

- Пользователи могут загружать и скачивать свои файлы с любого устройства.
- Пользователи могут делиться файлами и директориями с другими.
- Между всеми устройствами происходить автоматическая синхронизация.
- Можно редактировать файл офлайн, синхронизация произойдет при выходе в сеть.

Нефункциональные требования

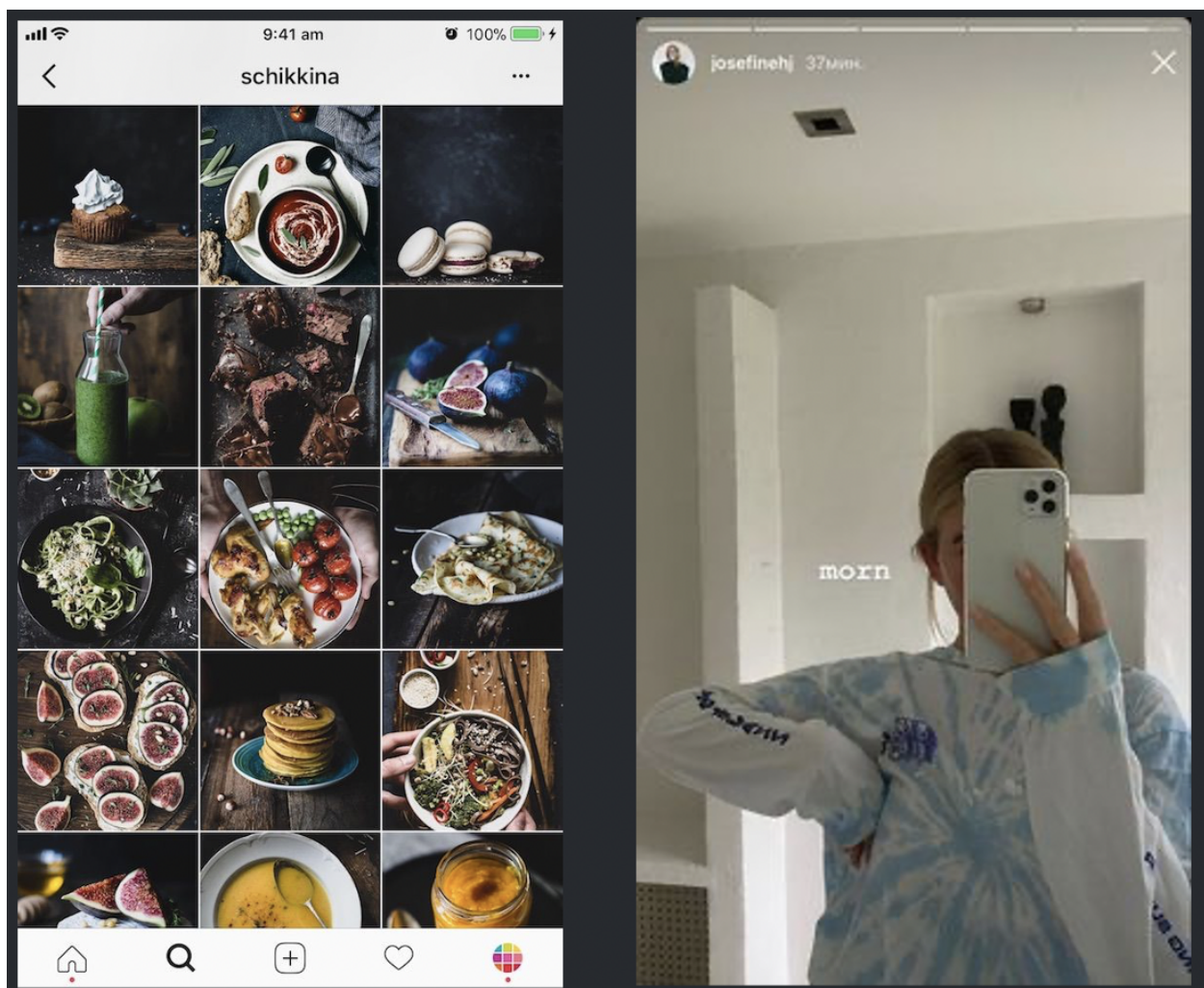
- (A) Атомарность – файл либо обновляется, либо нет, но не становится «битым».
- (C) Консистентность – при любом сценарии работы данные в облаке всегда валидны.

- (I) Изоляция – можно править разные файлы с разных устройств, все будет хорошо.
- (D) Устойчивость – после зеленой галочки на файле кофе уже можно выливать на ноутбук.

> Приложение для выкладывания фотографий

Зачем нужен такой сервис?

Чтобы делиться фотографиями в зеркале и с едой с друзьями.



Функциональные требования

- Пользователи могут загружать, скачивать и просматривать фотографии.
 - Пользователи могут искать другие фотографии по описанию.
 - Пользователи могут подписываться на фотографии друг друга.
 - Есть лента из популярных фото всех отслеживаемых пользователей.
-

Нефункциональные требования

- Высокая доступность сервиса – иначе пользователи перейдут в Snapchat или TikTok.
- Лента с фотографиями должна отображаться за сотни миллисекунд (иначе см п. 1).
- Консистентность не так критична – подписчики сколько-то потерпят без ваших селфи.
- Высокая надежность сервиса – ни одно селфи или фото еды не должно потеряться.

> Телеграм

Зачем нужен такой сервис?

Чтобы от общения вас отвлекали с работы, где не купили Slack.



Функциональные требования

- Можно общаться один на один в чатах с другими пользователями.
- Можно понять, онлайн ли пользователь и когда был последний раз.
- Сервис обеспечивает постоянное хранение всей истории переписки.

Нефункциональные требования

- Общение происходит в реальном времени с минимальными задержками.
- Консистентность – одинаковые сообщения в чатах на всех устройствах.
- Высокая доступность сервиса – иначе пользователи перейдут в там-там.

Бонус: стикеры, групповые чаты, кружочки с видео, беседы как в Clubhouse, стена.

> Твиттер

Зачем нужен такой сервис?

Чтобы делиться короткими сообщениями со всеми.



Функциональные требования

- Пользователи могут добавлять короткие посты.
- Посты могут содержать фото или видео.
- Пользователи могут отслеживать посты других.
- Есть лента из твитов всех отслеживаемых пользователей.

Нефункциональные требования

- Высокая доступность сервиса – иначе пользователи вернутся в живой журнал.
- Лента с твитами должна отображаться за сотни миллисекунд (иначе см п. 1).
- Консистентность не так критична – подписчики сколько-то потерпят без ваших мыслей.
- *Бонус*: поиск, реплаи, горячие темы, уведомления, рекомендации, аватарки с NFT обезьянами.

> Нетфликс

Зачем нужен такой сервис?

Чтобы смотреть любимые фильмы, сериалы и передачи.



Функциональные требования

- Пользователи могут просматривать фильмы и сериалы.
- Пользователи могут искать контент по названию.
- Сервис отслеживает оценки и статистику просмотров.
- Пользователям рекомендуются новые фильмы и сериалы.

Нефункциональные требования

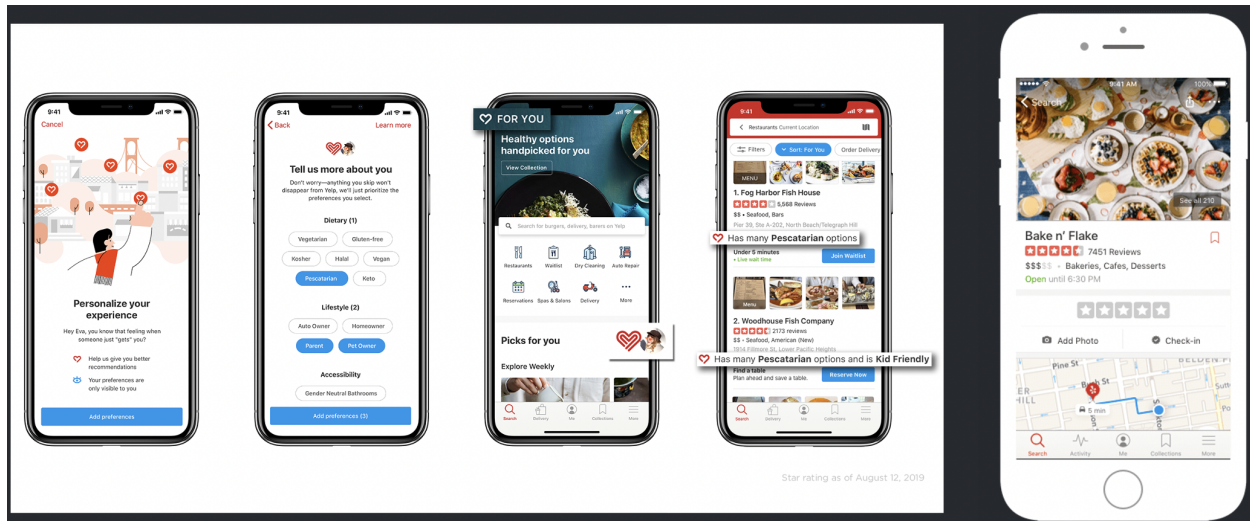
- Высокая доступность сервиса – иначе пользователи будут смотреть котиков у конкурентов.
- Высокая отзывчивость сервиса – видео должны проигрываться без подвисаний.
- Консистентность не так критична – после недельного ожидания одна минута не заметна.

Бонус: жанры, популярное, избранное, списки на посмотреть потом.

> Поиск ресторанов

Зачем нужен такой сервис?

Чтобы поесть не холодную еду из зеленых и желтых сумок.



Функциональные требования

- Пользователи могут добавлять/обновлять/удалять заведения.
- Пользователи могут находить подходящие заведения поблизости.
- Пользователи могут оставлять отзывы к посещенным местам с фото и оценками.

Нефункциональные требования

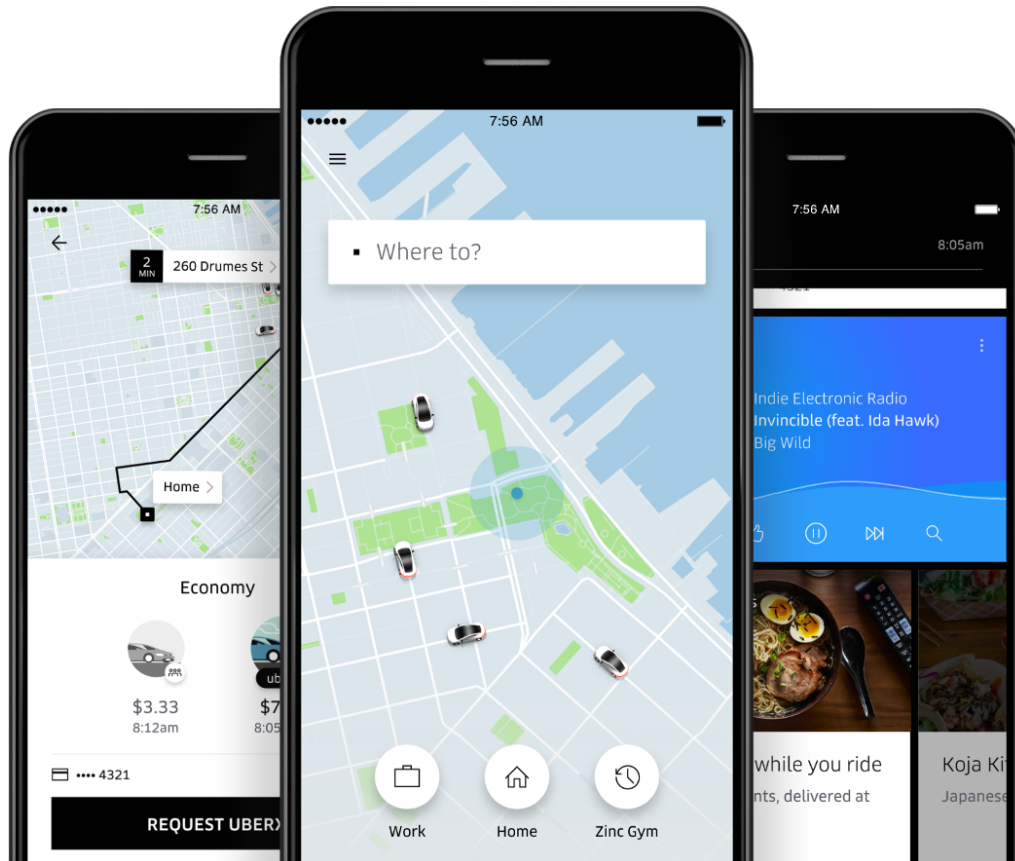
- Поиск должен происходить быстро, иначе пользователь закажет доставку.
- Гораздо большая нагрузка ожидается на подсервис поиска, чем на редактирование.

Бонус: фильтры по категориям, рекомендации новых заведений.

> Сервис такси

Зачем нужен такой сервис?

Чтобы легко доехать из точки А в точку Б с водителем.



Функциональные требования

- Водители могут выходить на смену и ждать заказ.
- Пассажиры могут находить водителей поблизости.
- Пассажиры могут заказывать поездку, для нее находится водитель.
- Позиции отслеживаются с момента принятия поездки и до ее окончания. В конце поездки водитель продолжает смену.

Нефункциональные требования

- Высокая доступность сервиса – иначе пассажиры поедут на метро, а водители на вокзал.
- Низкое время ожидания для пассажира, маленький простой для водителя.

Бонус: оценки водителям, оценки пассажирам, объяснение ценообразования.