# Оглавление

# Введение

Хотим сделать чат (почему?)

# Исследовательская часть

Есть куча прог, но в них либо

* только текст
* только графика в режиме совместного редактирования
* графика в режиме совместного редактирования с прилепленным сбоку текстовым чатом
* текстовый чат с пересылкой картинок, но без их совместного редактирования

При этом проги

* либо исключительно веб-приложения с невозможностью подключить обычную прогу-клиент в силу закрытости протокола
* либо имеют исключительно настольные клиенты, с той же проблемой закрытости (MSN)
* либо имеют открытый протокол, который фиг расширишь под наши нужды, и/или результат будет несовместим с существующими клиентами (сомнительный пункт, т.к. нужно реально изучать эти протоколы, чтобы утверждать такое; но звучит-то хорошо!)

Посему мы замахиваемся на

* свой протокол онлайн-конференции с функциями
  + передачи текста,
  + передачи только что нарисованных в проге изображений среди текстового потока,
  + коллективной работы над одним и тем же сообщением, при том что обычный текстовый чат никуда не пропадает
* свою прогу-сервер,
* свою прогу-клиент, причем в двух исполнениях – веб-приложение и настольное приложение (почему нет? ну, не сделаем для демо-версии, подумаешь)

Вот. (эх, как бы я хотел, чтобы РПЗ можно было реально писать вот в таком стиле!... как там… «ибо нефиг»)

Теперь это надо цивильно растянуть на пару страниц.

(На сколько именно?)

Нужно написать про причины выбора WebSockets как транспорта для нашего протокола (причины-то мы знаем – позволяет швырять данные туда-сюда без overhead’а полновесных HTTP-запросов; позволяет серверу связываться с клиентом, а не клиенту опрашивать сервер каждые N миллисекунд; легко реализуем как в веб-приложении (точнее, там он уже реализован браузером), так и в настольной проге (придется писать самому эту бредовую штуку с рандомными пробелами и цифрами, но это терпимо); хотя стандарт принадлежит группе HTML5 и еще находится в стадии черновика, и по-хорошему реализован в двух с половиной браузерах, может быть эмулирован в других браузерах посредством Flash-костыля)… так вот, причины-то мы знаем, но где про это писать, в исследовательской или конструкторской части?

Наверное в исследовательской, т.к., смотрим в методу,

В конструкторской части студент вырабатывает организационно-техническую структуру разрабатываемого программного средства. Представляет структурную схему и блок схему алгоритма функционирования, функциональную и программную структуру программного средства с указанием связей между модулями. Делает краткий вывод о составе и программной структуре разрабатываемого программного средства.

# Конструкторская часть

# Технологическая часть

# Заключение

# Приложение А

# Список литературы