

1. Прототипы экранных форм

1) Начало работы с ботом (команда /start)

После запуска данной команды бот переходит в активное состояние – если у пользователя уже есть подписки, то бот может присылать уведомления.

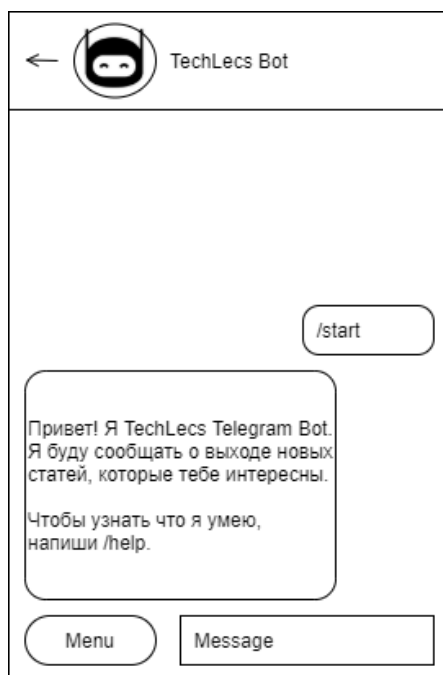


Рис. 1.1. Начало работы с ботом

2) Меню команд

Отображает доступные команды и их назначение. Данное меню аналогично выводу команды /help.

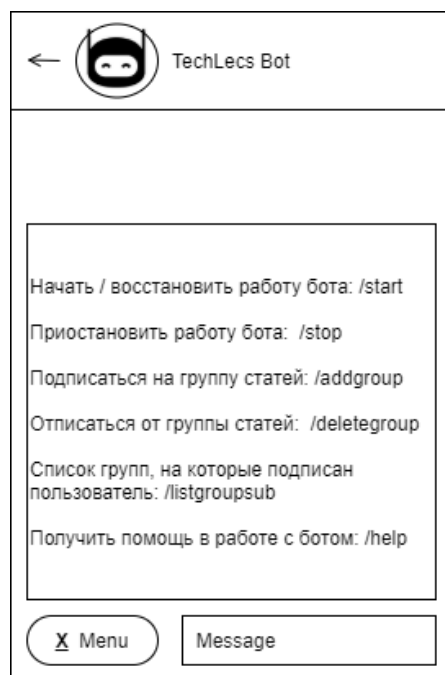


Рис. 1.2. Меню команд

3) Приостановка работы бота (команда /stop)

Бот не будет присылать уведомления. Все подписки пользователя сохраняются.



Рис. 1.3. Приостановка работы бота

4) Подписка на группу статей (команда /addGroup)

Если команда введена без ID, то будет показан список всех доступных групп. При вводе ID выполняется подписка на группу.



Рис. 1.4. Подписка на группу

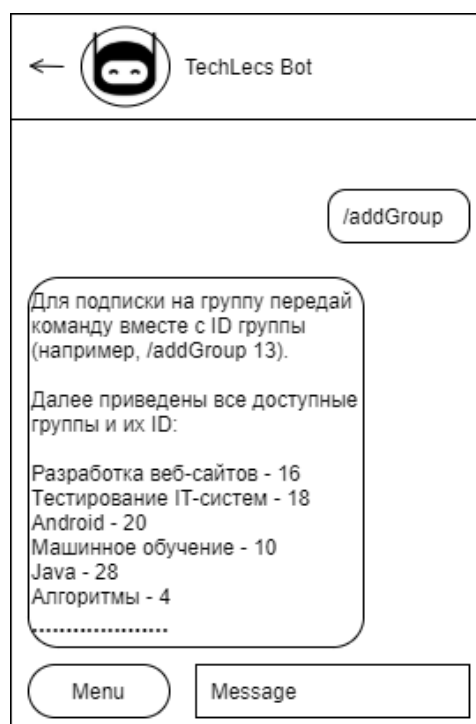


Рис. 1.5. Список доступных групп

5) Отписка от группы статей (команда /deleteGroup)

Если команда введена без ID, то будет показан список всех подписок пользователя. При вводе ID выполняется удаление подписки.

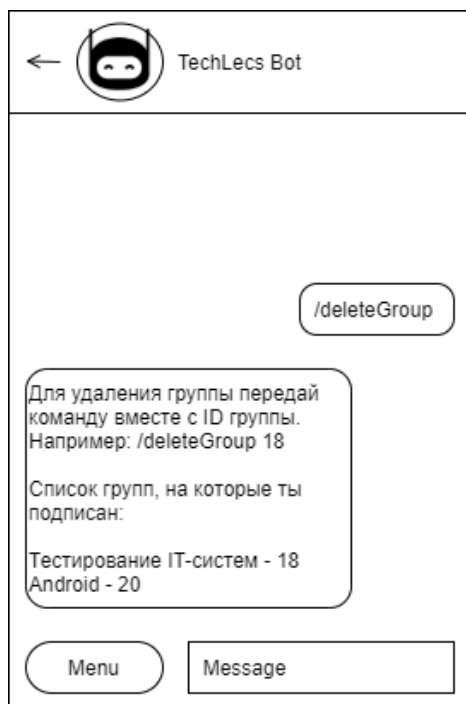


Рис. 1.6. Список подписок

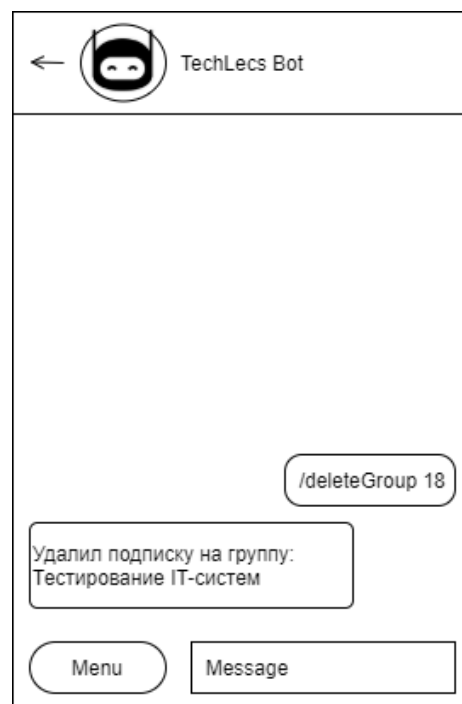


Рис. 1.7. Удаление подписки

6) Список групп пользователя (команда /listGroupSub)

Показывает на какие группы подписан пользователь. Бот проверяет наличие новых статей в данных группах и присылает уведомление при выходе новой статьи.

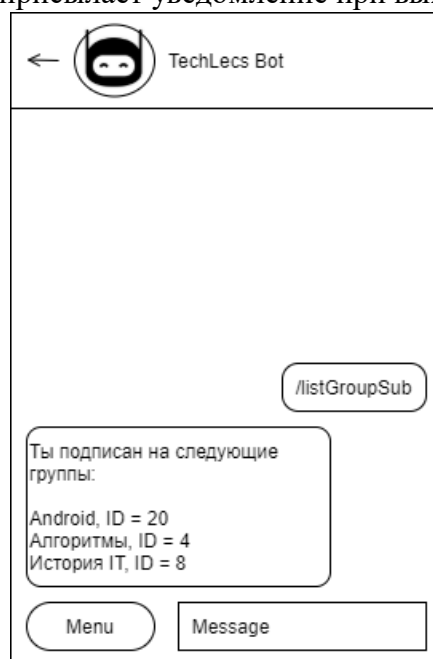


Рис. 1.8. Список подписок

7) Статистика работы бота

Получение разработчиком статистики использования бота — кол-во активных пользователей, на какие группы имеются подписки, на какие группы сделано больше всего подписок.



Рис. 1.9. Статистика работы бота

2. Диаграмма сущностей

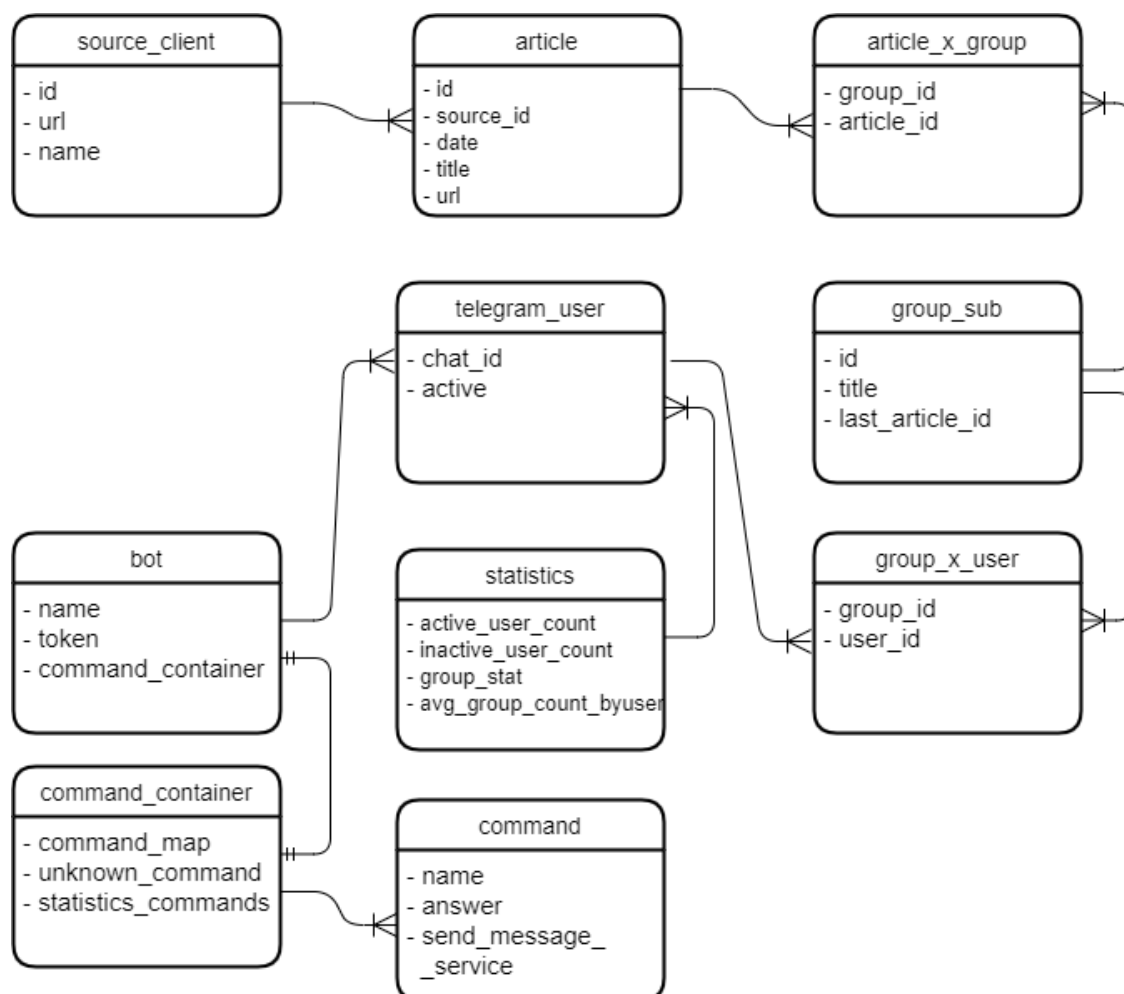


Рис. 2.1. Диаграмма сущностей

Source_client – сущности, используемые для связи с внешними ресурсами.

Article – сущность статьи, каждая статья может располагаться только на одном внешнем ресурсе

Group_sub – группы статей, на которые можно подписаться. В каждой группе может быть много статей, каждая статья может относиться к разным группам. Поэтому связь между article и group_sub многие-ко-многим.

Telegram_user – пользователь бота, связь с group_sub многие ко многим.

Bot – сущность бота, поля name и token необходимы для запуска бота в telegram. В поле command_container содержатся все доступные для выполнения команды. Создается только один экземпляр данной сущности.

Command_container – command_map содержит основные команды (имя команды(например, “/start”) и соответствующую сущность command). Statistics_commands – набор команд для получения статистики работы бота. Unknown command соответствует сущности command с неизвестной командой (используется в случае ввода несуществующей команды). Создается только один экземпляр сущности command_container.

Command - name – имя команды (например, “/start”), answer – сообщение, формируемое в качестве ответа на данную команду.

3. Разработка API системы

Табл. 3.1. API системы

Название функции	Описание действий	Входная информация	Выходная информация
Обработка событий ботом	Бот получил сообщение, должен на него ответить	Сообщение пользователя	Ответ бота в зависимости от типа входного сообщения
Запуск команды	Выполнение основной логики, требуемой определенной командой	Экземпляр сущности command.	Сообщение для ответа
Найти активных пользователей, подписанных на определенную группу	Получение списка пользователей, у которых бот находится в активном состоянии	Объект group_sub	Список, состоящий из объектов telegram_user
Найти неактивных пользователей, подписанных на определенную группу	Получение списка пользователей, у которых бот остановлен	Объект group_sub	Список, состоящий из объектов telegram_user
Удалить подписку	Удаляет из базы данных подписку определенного пользователя на определенную группу	Chat_id, объект group_sub	—

Продолжение табл. 3.1.

Название функции	Описание действий	Входная информация	Выходная информация
Подписаться на группу статей	Сохраняет в базе данных подписку определенного пользователя на определенную группу	Chat_id, объект group_sub	—
Получить подписки пользователя	Получение списка объектов group_sub – групп, на которые подписан пользователь с chat_id	Chat_id	Список объектов group_sub
Получить список групп	Получение списка групп определенного сайта	Объект source_client	Список объектов group_sub
Вычислить статистику работы бота	Вычисление кол-ва активных и неактивных пользователей, получение статистики по всем группам, вычисление среднего числа подписок на пользователя	—	Объект statistics с заполненными атрибутами
Найти новые статьи	Обращение на сайт, проверка выхода новых статей в заданной группе. Получение новых статей в случае их выхода.	Объект group_sub	Список объектов article
Отправить уведомление	Уведомление о выходе новых статей в определенной группе. Показывает пользователю описание статьи и ссылку на нее.	Объект group_sub, Список объектов articles	—

4. Иерархическая структура работ (ИСР)

1) Разработка технического задания

- 1.1) Определение требований к системе
- 1.2) Разработка прототипов форм
- 1.3) Определение сроков разработки

2) Создание телеграм бота

- 2.1) Создание бота, добавление его в проект
- 2.2) Заполнение меню с поддерживаемыми командами
- 2.3) Разработка функций обработки сообщений и команд

3) Разработка парсеров

- 3.1) Разработать парсер сайта Habr
- 3.2) Разработать парсер сайта JavaRush

3.3) Разработать парсер сайта DZone

- 4) Разработка серверной части
 - 4.1) Разработка базы данных
 - 4.2) Разработка функции получения списка групп определенного сайта
 - 4.3) Разработка функции получения списка подписок определенного пользователя
 - 4.4) Разработка функций остановки и возобновления работы бота
 - 4.5) Добавление возможности подписаться на группу статей
 - 4.6) Добавление возможности удалить подписку на группу статей
- 5) Разработка модуля для отправки уведомлений
 - 5.1) Разработка функций проверки выхода новых статей
 - 5.2) Разработка функции отправки уведомления
 - 5.3) Добавить планировщик задач, который должен запускать функции 5.1, 5.2 каждые 15 минут.
 - 5.4) Разработка функций сбора и вычисления статистики работы бота
- 6) Разработка тестов
 - 6.1) Ввод несуществующей команды
 - 6.2) Ввод сообщения, не являющегося командой
 - 6.3) Подписка на все возможные группы
 - 6.4) Удаление всех групп из подписок
 - 6.5) Тестирование уведомлений
5. Оценить время выполнения проекта по методу PERT.
 - Для оценки необходимо количественно оценить состав работ: указать кол-во сущностей, форм и методов арг
 - Для каждого вида элементов определить пессимистичные, оптимистичные и средние трудозатраты
 - Вывести общие трудозатраты проекта в чел. х мес.

Количество пользовательских экранов $N_{ui} = 7$

Количество обработчиков на экранах $N_{act} = 1$

Кол-во бизнес объектов $N_{BO} = 8$

Кол-во бизнес методов $N_{BM} = 11$

Кол-во обрабатываемых сайтов $N_s = 3$

Табл. 1. Пессимистичные, оптимистичные и средние трудозатраты

	Пессимистичные	Средние	Оптимистичные
Пользовательские экраны	3	2	1
Обработчики	16	7	5
Бизнес объекты	10	3	2
Бизнес методы	10	4	2
Парсеры сайтов	17	8	6

$$E_{ui} = \frac{3 + 4 * 2 + 1}{6} = 2$$

$$CKO_{ui} = 0.3$$

$$E_{act} = 8.2$$

$$CKO_{act} = 1.83$$

$$E_{BO} = 4$$

$$CKO_{BO} = 1.3$$

$$E_{BM} = 4.7$$

$$CKO_{act} = 1.3$$

$$E_s = 9.2$$

$$CKO_s = 1.83$$

$$106.2$$

$$E_{\Sigma} = \sum_i E_i * N_i = 2 * 7 + 8.2 + 4 * 8 + 4.7 * 11 + 9.2 * 3 \approx 133.5 \text{ ч. ч.}$$

$$CKO_{\Sigma} = \sqrt{\sum_i CKO_i^2 * N_i} \approx 6.8 \text{ ч. ч.}$$

$$E_{итог} = E_{\Sigma} + 2CKO_{\Sigma} = 147.1 \text{ ч. ч.}$$

$$E = 4E_{итог} = 588.4 \text{ ч. ч.}$$

$$E_{мес} = \frac{E}{132} \approx 4 \text{ мес}$$

Команда состоит из 2 человек, следовательно, $E_{мес} \approx 2 \text{ мес}$

6. Базовое расписание в виде диаграммы Ганта.

