

ТЗ для Чат-бота для регистрации гостей мероприятия

Версия 0.3

Основное назначение бота

- Регистрация участников проектов на мероприятие через кураторов и руководителей проектов
- Подтверждение участников проектов

Дополнительные данные

- БД бота: SQLite, mySQL (желательно)
- Внешняя БД с зарегистрированными гостями. Общение по Rest API. Два адреса:
 - Передача данных о гостях
 - Передача данных об авто
- Запущена виртуальная машина на РегПу. На машине параллельно запущен ещё один бот
- Установка нового бота не должна влиять на работу текущего.
- Репозиторий в Git: [SystoTGBot](#)
- Платформа разработки не имеет значения.

Определения

- Мероприятия - некоторое событие, которое определяется датой начала его проведения и временем окончания регистрации на него.
- Время окончания регистрации - дата и время, когда возможна регистрация участников на мероприятие. Дата и время определяется по Мск (+3). После наступления данного времени, регистрация, очевидно, невозможна
- Мероприятие считается будущим, если дата/время окончания регистрации больше текущего по Мск (+3).
- Администратор - пользователь, который осуществляет финальное подтверждение проекта и его участников.
- Куратор - пользователь, являющийся посредником между проектами и администраторами. Куратор может курировать до 30 проектов.
- Проект - некоторая команда с руководителем и участником, относящаяся к конкретному мероприятию.
- Руководитель проекта - отвечает за состав конкретного проекта и передает список на предварительное согласование куратору.

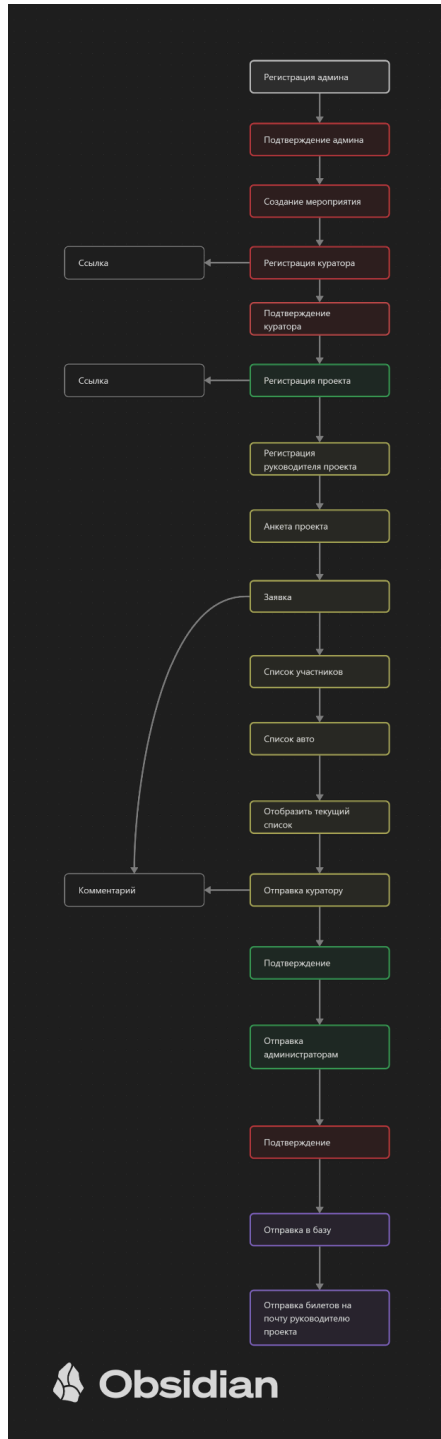
- Незарегистрированный пользователь, если его:
 - Нет в таблице пользователей;
 - На него нет зарегистрированных проектов для будущих мероприятий.

Роли

Роль	Доступные действия меню	Доступные действия чат
Незарегистрированный пользователь	<ul style="list-style-type: none"> • Нет 	<ul style="list-style-type: none"> • Регистрация как админа (секретная команда) • Заполнение данных как куратора (ответ на ссылку) • Ответ "Свяжитесь с ваших куратором или организатором мероприятия" (любой текст)
Администратор	<ul style="list-style-type: none"> • Создание мероприятия • Генерация ссылки на регистрацию куратора проектов • Проверка статуса любой заявки • Проверка нерассмотренных заявок 	<ul style="list-style-type: none"> • Подтверждение заявки на куратора (ответ на заполненную анкету куратора) • Подтверждение заявки на проекты • Подтверждение добавления участников
Куратор проектов	<ul style="list-style-type: none"> • Создание проекта • Генерация ссылки на регистрация участников • Добавление участников в проект списком • Просмотр состава проекта со статусом регистрации • Отправка заявки на регистрацию участников проекта • Проверка статуса заявки 	<ul style="list-style-type: none"> • Добавление участников проекта списком (ответ на любой текст)
Руководитель проекта	<ul style="list-style-type: none"> • Статус регистрации в проекте 	<ul style="list-style-type: none"> • Регистрация в проекте (Ответ на ссылку по регистрации) • Статус регистрации в проекте (Ответ на любой текст) • Получение pdf-файла со своим qr кодом для входа на мероприятие

Схема процессов

Сквозной процесс (overview)



Зависимость процессов и формы

Пользовательские сценарии

Незарегистрированный пользователь пишет в чат

1. Если в чат сообщение пишет незарегистрированный пользователь, то бот отображает сообщение “Для регистрации обратитесь к вашему куратору или организаторам мероприятий.” Данное сообщение не дублируется в той же сессии пользователя.
2. [Незарегистрированным](#) считается пользователь, если у него нет заявок на будущие мероприятия и нет определенной выше роли администратора или куратора.

/Restart Сброс состояния или процесса

Любому пользователю доступна через меню и в чате команда /Restart, которая останавливает любой процесс или модификатор, запущенный ранее. Позволяет начать любую доступную операцию заново.

/RegSystoAdmin Регистрация нового администратора с помощью секретной команды

1. Пользователь вводит команду "/RegSystoAdmin".
2. Бот просит ввести имя пользователя для регистрации (может отличаться от имени\названия аккаунта), предзаполняет текущим именем пользователя.
3. Доступные кнопки: “Отправить” и “Отмена”.
 - a. В случае нажатия “Отмена” процесс останавливается.
4. Пользователь нажимает кнопку “Отправить”.
5. В БД бота в таблицу пользователей добавляет запись с ролью админ (Role=1). Для новой записи устанавливается статус “На рассмотрении” (0).
6. Существующие админы в чате Бота получают сообщение “Запрос на регистрацию админа:”
[tgUserName] (номер аккаунта [tgId], имя аккаунта [tgAccountName]).
Подтвердить? Кнопки “Да” и “Нет”.
7. В случае нажатия “Да” для админа устанавливается статус “Подтверждён” (1)
8. В случае нажатия “Нет”
 - a. Бот проверяет состояние нового пользователя.

- i. Если статус равен “Подтверждён”, то бот выводит сообщение, что данный администратор уже подтверждён. Вы уверены, что хотите отменить заявку и удалить нового админа. Кнопки “Да, уверен!” и “Нет, оставь!”.
 - 1. Если нажата кнопка “Да, уверен!”, то данные админа удаляются из таблицы.
- ii. Если статус равен “На рассмотрении”, то данные админа удаляются из таблицы.

/AddEvent Создание мероприятия

1. Добавление мероприятия возможно только администратором
2. Добавление мероприятия осуществляется с помощью меню “Добавить мероприятие” или команда /AddEvent
3. После выбора в меню “Добавить мероприятие” администратор вводит название мероприятия, дату/время прекращения регистрации гостей.
 - a. Дата и время определяются по Москве
4. Мероприятие добавляется в таблицу с указанными данными.
5. Редактирование мероприятия доступно через БД

Примечание: уточнить у разработчика, как подключиться к БД для редактирования записей через SQL, нужно ли блокировать БД на время редактирования, как?

/ListEvents Просмотр списка мероприятия

1. Просмотр списка доступен только администраторам
2. Осуществляется с помощью меню “Мероприятия” или команда /ListEvents
3. Бот отображает: “Список будущих и активных мероприятий:”
4. Для каждого мероприятия отображается: [EventName] (Время окончания регистрации: [EventRegEndTime])

/RegCourator Генерация ссылки на регистрацию куратора проектов

1. Для администратора доступен пункт меню “Регистрация куратора” (команда /RegCourator)
2. Бот формирует сообщение формата “Вам предложено зарегистрироваться как куратор. Перейдите по ссылке для заполнения анкеты”. Для слова анкета сформирована ссылка формата “https://t.me/<юзернейм_бота>?regcurator=id”, где tgID администратора, который сгенерировал ссылку.
3. Данное сообщение форвардится средствами ТГ
4. Пользователь может указать одного или несколько пользователей ТГ

5. Другой пользователь (кандидат) получает данное сообщение
6. При переходе по ссылке Бот, проверяет, есть ли в БД бота пользователь с ролью куратор (Role = 2)
 - a. Если в таблице есть такой пользователь, то отображается сообщение “Вы уже зарегистрированы”. Процесс для куратора прекращается.
7. Если нет, то бот отображает анкету, в которой заполняется:
 - a. Фамилия и имя куратора
 - b. Адрес почты
8. После заполнения доступно действие.
 - a. Отправить Анкету?
 - b. Кнопки “Да”, “Нет”.
9. При нажатии кнопки “Нет”, бот отображает сообщение “Для повторного создания анкеты позже повторно перейдите по ссылке для регистрации куратора”.
10. После нажатия “Да” Бот формирует запись в таблице пользователей
 - a. Номер аккаунта: tgID
 - b. Имя, Фамилия: UserName
 - c. Имя аккаунта (базовое): tgAccountName
 - d. Роль = 2
 - e. Признак рассмотрения = 0
 - f. Id администратора, который сгенерировал ссылку.
11. Администраторы получают сообщение:
 - a. Заявка на добавление куратора:
 - i. Имя аккаунта
 - ii. Указанные Фамилия и Имя
 - iii. Accountid
 - iv. Почта
 - b. Подтвердить добавление куратора?
 - i. Да: Признак рассмотрения = 1
 1. Сообщение: Куратор успешно добавлен.
 - ii. Нет: Проверка, аналогичная добавлению администратора.

/RegNewProjectLink Регистрация проекта

1. Генерация ссылки на регистрацию проекта доступна пользователю с ролью “Куратор” с помощью меню или команды /RegNewProjectLink
2. Бот определяет количество будущих мероприятий:
 - a. Если в системе нет будущих мероприятий, то Бот отображает оповещение “Будущих мероприятий не найдено, обратитесь к организаторам для добавления”
 - b. Если в БД найдено одно будущее мероприятие, то бот запоминает его идентификатор для дальнейшего использования.
 - c. Если в БД найдено более одного будущего мероприятия, то бот предлагает выбрать один из них, отображая кнопки с Датой начала мероприятия и его

названием. Выбранное пользователем мероприятие бот запоминает в виде идентификатора.

3. Бот генерит ссылку для регистрации проекта. Ссылка (или связанный с ней объект в БД) должна содержать информацию о мероприятии и кураторе, который сгенерил данную ссылку.
4. Ссылка пересылается руководителям проектов средствами ТГ.
5. При переходе руководителя проекта по ссылке, бот проверяет наличие зарегистрированных проектов данного пользователя для данного мероприятия.
 - a. Если такие есть, то бот отображается текст “У вас уже есть зарегистрированный проект на данное мероприятие.”
 - Отображаются список проектов с кнопками
 - “Отобразить список участников?” - см. /GetPartyList
 - “Добавить участников?” - см. /AddParticipantsText
 - “Добавить авто?” - см. /AddAutoText
6. Если для пользователя нет зарегистрированных проектов, то бот предлагает ввести последовательно следующие данные, отобразив при этом часть системных данных “Зарегистрируйте новый проект на мероприятие: [EventName]:
 - a. Ваше имя и фамилия: [UserName] (предзаполняется данными пользователя, если пользователь есть в БД)
 - b. Ваш адрес электронной почты (на данный адрес будут отправлены билеты участников): [eMail] (предзаполняется данными пользователя, если пользователь есть в БД)
 - c. Название проекта: [ProjectName]
 - d. Кнопка “Добавить?”.
 - Если данного пользователя нет в БД, то бот записывает нового пользователя в БД с параметрами [tgid], [tgAccountName], [UserName], [eMail], [Role]=3, [статус] = 0
 - Если пользователь есть в БД, то бот сверяет имя и почту пользователя.
 - Если данные (любое из полей) отличается, то бот запрашивает, “Необходимо ли изменить имя и почту? Обратите внимание, что данное действие будет применено ко всем зарегистрированным от вас проектам
 - Кнопки: Да, Нет?
 - Да, перезаписывает данные.
 - Нет, сохраняет данные в БД без изменений.
 - Записывает новый проект в БД с параметрами [id], [idEvent], [idCourator], [idProjectLead], [статус] = 0
 - Формирует новую заявку с параметрами [id], [idProject], [limit] = #defaultLimit, [limitAuto] = #defaultLimitAuto, [статус] = 0
 - Бот проверяет, есть ли текущий пользователь в заявках на текущее мероприятие:
 - Если такой пользователь существует, то бот завершает процесс.

4. Далее бот проверяет наличие открытых заявок для проекта:
 - a. Если у руководителя есть заявки со статусом 1 или 2, для данного проекта то бот отображает сообщение “У вас есть заявки, переданные вашему куратору или администратору. Дождитесь окончания их рассмотрения!”
 - b. Если у руководителя есть заявки со статусом 0, то бот проверяет количество гостей в такой заявке и отображает сообщение.
 - i. Если количество гостей равно лимиту, то бот отображает сообщение “Вы больше не можете добавить гостей. Отправить заявку на рассмотрение вашему куратору?”.
 1. Да - см. Отправка заявки куратору
 2. Нет - процесс прекращается.
 - c. Если у руководителя проекта нет активных заявок на будущие мероприятия для его проектов, бот формирует новую заявку с параметрами [id], [idProject], [limit] = #defaultLimit, [limitAuto] = #defaultLimitAuto, [State] = 0. [Id] запоминается для занесения участников.
5. Далее бот отображает сообщение: “Введите участников списком в формате Имя и Фамилия, где каждый участник представлен отдельной строкой. Также можно ввести имена участников по-очереди, отправляя в чат по-одному. В таком случае для завершения ввода введите команду /cancel, выберите в меню “Ввод завершён” или откажитесь кнопкой “Нет” от продолжения ввода”.
6. После получения сообщения бот пытается обработать список построчно:
 - a. Для каждой строки бот формирует новую запись в таблице участников с параметрами: [idRequest] = номер активной заявки, к которой привязывается гость, [GuestNameAutoNumber] = Введённое имя, [Type] = 1, [Role] = “Гость”, [Status] = 0.
 - b. После добавления гостя в таблицу бот сверяет количество гостей в активной заявке с лимитом. Если количество гостей в заявке со статусом 0 равно количеству в поле “limit” - бот выводит сообщение “Достигнут лимит X=[limit] в количестве гостей. Отправьте заявку на подтверждение куратору”. Обработка списка или ожидание ввода следующего гостя прекращается.
 - c. Если была введена только одна строка и лимит не достигнут, бот выводит сообщение “Добавить другого участника?” “Да”, “Нет”.
 - i. Если “Да”, бот ожидает ввода нового сообщения.
 - ii. Если “Нет”, бот прекращает обработку списка.
 - d. Если была введена команда /cancel или аналогичный элемент меню, то бот прекращает обработку списка.
 - e. Если было в сообщении было обработано несколько строк, то после их окончания бот прекращает обработку списка.
7. После обработки списка бот отображает сообщение:
 - a. Мероприятие: Дата, Наименование
 - b. Проект: Название проекта
 - c. В проект успешно добавлены следующие участники:
 - i. [idучастника1]: Участник 1.
 - ii. [idучастника2]: Участник N.

8. После ввода сообщения бот выводит сообщение: “Проверьте, список и убедитесь, что список корректен”. Кнопки:
 - a. Да, всё верно! - бот прекращает ввод гостей по заявке. Отправить заявку куратору? Кнопки “Да?”, “Нет?”.
 - i. Да - см. Отправка заявки куратору
 - ii. Нет. - процесс останавливается.
 - b. Нет, очистить список. Бот очищает список гостей для данной заявки.
9. Команда может быть повторена до достижения лимита участников в заявке.

/AddAutoText Добавление авто в проект списком

1. Руководитель проекта может добавить список авто с помощью команды /AddAutoText или выбрав в меню “Добавить в проект авто”. Данная команда доступна, если у руководителя проекта есть активные проекты на будущие мероприятия.
2. Если для заявки установлено количество [limitauto] = 0, то добавить авто по заявке нельзя. Бот отображает сообщения - в данной заявке авто не предусмотрены.
3. Действия по добавлению авто аналогичны списку гостей. Разница заключается в том, что при добавлении записи в таблицу устанавливается тип Type=2.
4. Действия может быть повторено до достижения количество записей для заявки равного значению атрибута [limitauto].
 - a. Если для заявки не установлено количество [limitauto] = NULL, то ограничение по количеству авто не применяются.

/GetPartyList Просмотр состава проекта со статусом регистрации

1. Куратору и руководителю проекта доступна в меню и в чате команда /getpartylist.
2. После ввода команды бот проверяет количество активных проектов куратора (проектов, привязанных к мероприятиям с открытой регистрацией) или руководителя проекта.
 - a. Если проектов более одного, бот отображает список в формате:
 - i. Мероприятие1: Дата начала, Наименование, дата и время окончания регистрации.
 1. Проект 1: Наименование 1
 2. Проект 2: Наименование 2
 3. Для каждого проекта указывается кнопка “Выбрать”
 - ii. После выбора бот запоминает [id] проекта.
 - b. Если проект один, то бот запоминает [id] проекта без выбора.
3. Для выбранного проекта бот отображает вопросы:
 - a. Показать список всех участников?
 - b. Показать незарегистрированных участников?

- c. Отмена? - процесс останавливается.
- 4. После выбора бот отображает список участников в формате:
 - a. Проект: [Название проекта] ([Руководитель проекта])
 - b. Номер заявки: [Request.id]
 - c. Список гостей:
 - i. [Номер гостя]: [Фамилия Имя] ([Статус заявки])
 - d. Список авто:
 - i. [Номер авто]: [Фамилия Имя] ([Статус заявки])
- 5. Список отфильтрован в зависимости от выбора.
- 6. Если в списке есть хотя бы один незарегистрированный или авто участник (статус заявки = 0), бот предлагает “Отправить заявку на подтверждение?” (см. процесс ниже).
- 7. Если в отфильтрованном списке гостей или авто нет участников бот отображает сообщение “Список пуст”.

/SendPartyRequest Отправка заявки на подтверждение

- 1. Куратору или руководителю проекта доступна команда меню и в чате /sendpartyrequest
- 2. Бот проверяет наличие не отправленных заявок:
 - a. Для руководителя определяются заявки со статусом (состоянием) = 0 и внесённых участников или авто со статусом = 0.
 - b. Для куратора определяются заявки со статусом (состоянием) = 1 и внесённых участников или авто со статусом = 0.
- 3. Если в БД для пользователя обнаружено заявок более, чем на один проект, то бот выводит список активных проектов куратора (проектов, привязанных к мероприятиям с открытой регистрацией):
 - a. Мероприятие1: Дата начала, Наименование, дата и время окончания регистрации.
 - i. Проект 1: Наименование 1
 - ii. Проект 2: Наименование 2
 - iii. Для каждого проекта указывается кнопка “Выбрать”
 - b. После выбора бот запоминает [id] проекта.
 - c. Если проект один, то бот запоминает [id] проекта без выбора.
- 4. После выбора проекта, бот отображает данные:
 - a. Мероприятие1: Дата начала, Наименование, дата и время окончания регистрации.
 - b. Проект 1: Наименование 1
 - c. Номер заявки: [Номер заявки] (статус заявки)
 - d. Участники:
 - i. [Номер участника]: [Фамилия Имя] [[Роль]] ([Статус рассмотрения])
 - e. Авто:
 - i. [Номер авто]: [Название авто] ([Статус рассмотрения])
 - f. Список участников отображается только для неотправленных заявок.

- g. Если в соответствующем списке нет участников или авто, то в списке отображается “Список пуст”.
- h. После списка бот отображает вопрос: “Отправить список на подтверждение?”
 - i. Да. - См. ниже.
 - ii. Нет. - процесс останавливается
- 5. Если пользователь выбрал “Да”, бот отображает сообщение “Комментарий для заявки” и ожидает ввода текста.
 - a. Комментарий может отсутствовать (быть пустым).
- 6. После ввода комментария бот:
 - a. Меняет статус заявки на:
 - i. 1, в случае отправки куратору
 - ii. 2, в случае отправки администратору
 - b. Для всех гостей и авто из заявки устанавливается статус = 0.

Подтверждение куратором

- 7. Если заявка получила статус 1, то бот определяет куратора данного проекта и в чате *куратора* отображает сообщение:
 - a. Запрос на добавление участников. Заявок: XX (количество гостей для заявки со статусом “На рассмотрении”). Зарегистрировано ранее: YY (количество запросов для проекта со статусом заявки “Подтверждено”)
- 8. “Новая заявка на добавление гостей:
 - a. Мероприятие: Дата, Наименование, Дата и время окончания регистрации
 - b. Проект: Название, Руководитель проекта (почта)
 - c. Заявка: [номер заявки]
 - d. Комментарий: [Комментарий из заявки]
 - e. Гости в заявке: XX (зарегистрировано ранее - YY)
 - f. Авто в заявке: AA(новых) (зарегистрировано ранее - ZZ)
 - g. Участники:
 - i. [номер участника]: Имя [[Роль]]
 - ii. ...
 - h. Авто:
 - i. [номер авто]: Название и номер
 - ii. ...
 - i. Если в заявке только один гость или авто, то бот автоматически переходит в режим подтверждения по одному. См. ниже.
 - j. Кнопки для режима с несколькими гостями или авто:
 - i. “Подтвердить всех?” - Отображается, если более одного гостя или машины, см. ниже
 - ii. “Подтвердить по одному?” - Отображается, если более одного гостя или машины, см. ниже. Ниже.
 - iii. “Отказать всем?” - см. ниже

- iv. “Отмена” - процесс прекращается. Повторить можно командой /CheckOpenRequests
9. Если пользователь выбрал Подтвердить всех, то заявка получает статус 2.
10. Если пользователь выбрал Подтвердить по одному, то:
- a. Бот по очереди для каждого участника отображает:
 - i. Гость: [номер гостя], Имя [[Роль]]
 - b. Для каждой записи отображаются кнопки:
 - i. Подтвердить? - для гостя устанавливается статус = 1
 - ii. Отказать? - гость удаляется из списка.
 - iii. Отмена? - процесс останавливается. Повторить можно командой /CheckOpenRequests
 - c. После окончания списка гостей бот для каждого авто отображает
 - i. Авто: [номер авто], Название и номер
 - d. После окончания списка гостей или авто бот для данной заявки проверяет количество подтверждённых гостей или авто:
 - i. Если таких больше или равно 1, то заявка получает статус = 2, а все гости и авто получают статус = 0.
 - ii. Если гостей или авто нет, то заявка получает статус 9.

Подтверждение проекта

11. Если заявка получила статус = 2 (перешла на подтверждение администратору), а сама заявка относится к неподтвержденному проекту, то бот отображает в чате для администраторов запрос:
- a. “Запрос на добавление проекта:”
 - b. Мероприятие: [Название мероприятия]
 - c. Название проекта: [НазваниеПроекта]
 - d. Куратор: ФИО куратора, логин, почта
 - e. Подтвердите добавление проекта (кнопки):
 - i. Подтвердить. - См. ниже.
 - ii. Отклонить. - См. ниже
 - iii. Позже. - Остановить процесс
 - f. После нажатия “Подтвердить”, бот проверяет статус подтверждения проекта.
 - i. Если статус “Не рассмотрен”, то бот меняет статус на подтверждён = 1. Кем был подтверждён [Approved] устанавливается равным id текущего админа.
 - ii. Если статус уже установлен = “Подтверждён”, то статус не меняется.
 - iii. Если статус = “Отклонён”, то бот отображает сообщение, что проект был отклонён, администратор - такой. Подтвердите изменение решения или свяжитесь в администратором.
 - 1. Да. - Статус меняется на подтверждён. Кем изменён меняется на текущего админа. Предыдущий админ запоминается ботом.

2. Свяжусь с админом. - Статус не меняется. Процесс прекращается.
- g. После нажатия кнопки "Отклонить", бот проверяет текущий статус:
 - i. Если статус = Нерассмотрен, то меняется на Отклонить = 2 и кем изменён = Текущий админ.
 - ii. Если статус "Подтверждён", то сообщение, что статуса уже был подтверждён админом, перезаписать?
 1. Да. - см. ниже.
 2. Нет. - Переход на подтверждение участников.
 3. Позже. Процесс останавливается
 - iii. Если "Да", то статус меняется на подтверждён, меняется админ, предыдущий админ запоминается.
- h. В случае смены статуса предыдущий админ, который ставил статус, получает сообщение - проект был изменён. Новый статус = такой-то, админ - такой-то.
- i. В случае подтверждение проекта все админы получают сообщение:
 - i. Проект такой-то (карточка проекта с куратором и руководителем) был утвержден, таким-то админом.

Подтверждение регистрации администратором

12. Если проект подтверждён, то бот отображает список участников проекта для подтверждения.
13. Процесс подтверждения аналогичный процессу куратора, с тем отличием, что сообщение получают все администраторы.
14. Если какой-то администратор начал подтверждать состав, то остальные администраторы не имеют такой возможности, они получают оповещение о рассмотрении участников или авто другим администратором.

Оповещение о проведённом подтверждении

15. После того, как заявка полностью обработана (в заявке не осталось нерассмотренных участников или авто):
 - a. Заявка получает статус 9
16. Бот в чате администраторов отображает сообщения:
 - a. Обработана заявка [номер заявки]
 - b. Мероприятие: [Название мероприятия]
 - c. Дата мероприятия: [Дата мероприятия]
 - d. Проект: Название, куратор, руководитель
 - e. Подтвердил: Имя администратора
 - f. Подтвержденные участники (X - подтверждено в заявке):
 - i. Имя, роль
 - g. Подтвержденные авто (Y - по):
 - i. Имя, роль

- h. Отказано:
 - i. Имя
 - ii. ...
 - i. Всего в проекте подтверждено:
 - i. Участников: X
 - ii. Авто: Y
17. Аналогичное сообщение отображается в чате для куратора и руководителя проекта, для которого подтверждена заявка.

Выдача билетов

18. Для заявки со статусом 9 и TicketsSent = 0 бот проверяет количество участников со статусом “Подтверждено”.
- a. Если таких один и более, то бот формирует REST запрос в сторону билетной базы.
 - b. Состав запроса:
 - i. Название проекта
 - ii. Почта
 - iii. Список гостей
 - iv. Комментарий.
19. Для заявки со статусом 9 и TicketsSent = 0 бот проверяет количество авто со статусом “Подтверждено”.
- a. Если таких один и более, то бот формирует REST запрос в сторону билетной базы для записи авто.
20. После выдачи билетов:
- a. Для заявки устанавливается значение TicketsSent = 1
 - b. Администраторы, куратор и руководитель проекта получают сообщение в чате “Билеты отправлены на почту!”.

Проверка открытых заявок

21. Если после обработки проекта в списке заявок остались открытые, то бот отображает вопрос “На рассмотрении есть открытые заявки, показать?”
- a. Да? Начинается процесс работы с открытыми заявками (см. ниже).
 - b. Нет. Процесс прекращается.
22. Важно: Пока есть рассмотренные заявки, они первые отображаются в действиях (меню) админа.

/CheckOpenRequests Проверка нерассмотренных заявок

1. Администратору и куратору доступна команда “/CheckOpenRequests” и в меню “Отобразить не рассмотренные (открытые) заявки”.
2. Заявки группируются по проектам и мероприятиям.

3. Если есть открытые заявки для нескольких проектов, то бот отображает:
 - a. Мероприятие: Дата начала, название мероприятия (дата и время окончания)
 - b. Проекты:
 - i. Проект 1, Куратор, Руководитель проекта
 - ii. Проект 2, Куратор, Руководитель проекта
 - iii. Для каждого проекта отображается кнопка “Выбрать”.
4. После выбора проекта система показывает в чате список нерассмотренных запросов по гостям и по авто:
 - a. Если запрос в единственном числе, то отображается в режиме подтверждения по одному.
 - b. Если запросов более одного, то отображаются кнопки группового подтверждения:
 - i. Подтвердить всех? - групповое подтверждение.
 - ii. Подтверждать по одному? - Переход к подтверждению по одному.
 - iii. Отклонить всех
 - iv. Отменить.
5. Действия кнопок аналогичны сценарию “Подтверждение регистрации”.
6. Если подтверждены запросы для хотя бы одного гостя или авто, то система отправляет запрос в Базу билетов для их отправки руководителю проекта.
7. После отправки билетов бот переходит к процессу проверки открытых заявок и отображает запрос “На рассмотрении есть открытые заявки, показать?”:
 - a. Да? - Начинается процесс работы с открытыми заявками (см. ниже).
 - b. Нет. - Процесс прекращается.

/CheckRequestStatus Проверка статуса заявки

1. Администратору доступна команда “/CheckRequestStatus” и в меню “Проверить статус заявки”.
2. Бот предлагает ввести цифровой идентификатор [id] заявки (объект Request).
3. После ввода идентификатора система отображает карточку регистрации, аналогично запросу:
 - a. “Заявка: [id]
 - b. Мероприятие: [Название мероприятия]
 - c. Проект:
 - d. Руководитель:
 - e. Почта:
 - f. Куратор проекта:
 - g. Статус: [Статус заявки].
 - h. Установил: [имя администратора]
 - i. Дата установки: [дата изменения]
 - j. Если в заявке есть непросмотренные запросы, то добавляется кнопка “Показать нерассмотренные запросы из заявки?”

- i. Да - Переход к подтверждению заявки.
 - ii. Нет - Процесс прекращается.
 - k. Если заявка рассмотрена полностью, то отображается запрос “Показать состав заявки и статус рассмотрения запросов?”
 - i. Да - Отображается структура запросов и их статус (см. ниже)
 - ii. Нет - процесс прекращается.
- 4. Если пользователь выбрал команду посмотреть структуру запросов, то отображается:
 - a. Запросы на участников (отображается если запросов больше 0):
 - i. Все запросов: X
 - ii. Подтверждено: X1
 - iii. Отклонено: X2
 - b. Состав участников:
 - i. [Запрос1]: Имя, роль, статус
 - ii. [Запрос2]: Имя, роль, статус
 - c. Запросы на авто (отображается если запросов больше 0):
 - i. Все запросов: Y
 - ii. Подтверждено: Y1
 - iii. Отклонено: Y2
 - d. Состав авто:
 - i. [Запрос1]: Номер, статус рассмотрения
 - ii. [Запрос2]: Номер, статус рассмотрения
- 5. Если номер заявки не найден в БД, то бот отображает текст ошибки: “Заявка с номером [id] не найдена”.
- 6. Также команда доступа куратору проекта. Отличие в том, что для куратора бот дополнительно проверяет, что заявка относится именно к проекту куратора. В противном случае бот выдаёт ошибку поиска заявки.

/SendQuickRequestLink Отправка быстрой заявки

- 1. У администратора есть возможность отправить быструю заявку с автоподтверждением командой /SendQuickRequestLink или выбрав в меню “Отправить быструю заявку на регистрацию”
- 2. Бот спрашивает:
 - a. Количество гостей
 - b. Количество авто
 - c. Бот проверяет, чтобы количество не было больше чем количество в соответствующих лимитах по-умолчанию (настройки бота)
 - i. Если какое-то из количеств превышает настройки, то бот отображает соответствующее сообщение с ошибкой.
- 3. Бот формирует в базе запись для запроса без ссылки на проекта, но ссылкой на текущего администратора с параметрами:
 - a. IdProject (int) = 0

- b. idAdmin (int) = Идентификатор администратора
 - c. Limit (int) = введённое количество для гостей
 - d. LimitAuto (int) = введённое количество для авто
 - e. Type (int) = 1
4. Бот формирует ссылку с указанием параметра Id = код заявки.
 5. Ссылка может быть переслана средствами ТГ
 6. При открытии данной ссылки бот проверяет, что
 - a. Текущий пользователь НЕ является администратором
 - b. Текущий пользователь НЕ является куратором
 - c. Если проверка не пройдена бот отображает сообщение: “Данная заявка не может быть заполнена администратором или куратором”.
 7. Бот запрашивает у пользователя
 - a. ФИО
 - b. Почта
 8. После ввода данных бот создаёт запись в таблицу Пользователи с параметрами:
 - a. idTG = номер аккаунта в ТГ
 - b. tgAccountName = название аккаунта при регистрации (ака @account)
 - c. UserName = Наименование, введенное пользователем
 - d. eMail = почта пользователя
 - e. Role = 0
 - f. Status = 3 (автоподтверждён)
 9. После ввода данных бот создаёт запись в таблицу Проекты с параметрами:
 - a. idEvent = ссылка на ближайшее мероприятие
 - b. idProjectLead = ссылка на текущего пользователя
 - c. Status = 3 (автоподтверждён)
 - d. Approved (int) = ссылка на администратора
 10. Бот обновляет запись заявки ссылкой на созданный проект.
 11. Далее бот переходит в режим добавления гостей, аналогично запросу /AddParticipantsText Добавление участников в проект списком, если для заявки лимит гостей установлен больше 0.
 12. После ввода гостей бот переходит в режим добавления авто, аналогично запросу /AddAutoText, если для заявки лимит авто установлен больше 0.
 13. После окончания ввода гостей и авто бот предлагает отобразить состав заявки для проверки.
 14. После отображения состава заявки бот предлагает отправить заявку администратору.
 15. После отправки администратору, если заявка имеет тип автоподтверждения, то она получает этап окончания рассмотрения (настройка).
 16. Все запросы в заявке получают статус 3 (автоподтверждение).
 17. Бот формирует запрос в базу билетов на их выдачу для гостей и авто.
 18. Все запросы получают TicketsSent = 1 после успешного ответа билетной базы.

Объектная модель

Мероприятия / Events

Атрибуты

Id = идентификатор-инкремент

EventName = Название мероприятия

EventStartDate = Дата начала мероприятия

EventRegEndDateTime = Дата и время окончания регистрации

Пользователь / Users

Атрибуты

Id = идентификатор-инкремент

idTG = номер аккаунта в ТГ

tgAccountName = название аккаунта при регистрации (ака @account)

UserName = Наименование, введенное пользователем

eMail = почта пользователя

Role = Роль пользователя

Status = Состояние рассмотрения пользователя

Роли / Role

1 = Администратор

2 = Куратор

3 = Руководитель проекта

0 = Быстрый проект

Состояние

0 = создано, на рассмотрении

1 = подтверждено

[2 нет, поскольку такие пользователи не сохраняются]

3 = Автоподтверждение

Проект / Projects

Атрибуты

Id = инкремент
idEvent = ссылка на мероприятие
idCourator = ссылка на куратора
idProjectLead = ссылка на руководителя проекта
Status = состояние рассмотрения проекта
Approved (int) = кем подтверждён роль

Состояние

0 = новый, проект не подтверждён
1 = проект подтверждён
2 = отказ
3 = автоподтверждён

Заявка / Requests

Заявка требуется для контроля количества участников в одной подаваемой заявке.

Атрибуты

Id (inc) = идентификатор заявки, инкремент
IdProject (int) = идентификатор проекта
Limit (int) = ограничение участников
LimitAuto (int) = ограничение на автомобили
Type (int) = тип заявки, нужен для автоподтверждения
Comment (text) = комментарий к заявке
TicketsSent (int) = билеты отправлены
Stage (int) = этап рассмотрения заявки

Этап рассмотрения

0 = новая, на рассмотрении у руководителя проекта
1 = на рассмотрении у куратора
2 = на рассмотрении у администратора
9 = процесс рассмотрения завершён [настройка, можно будет двигать]

Типы

0 = Обычная
1 = Автоподтверждение

Участник / Participants

Атрибуты

Id = уникальный идентификатор гостя, инкремент
idRequest = номер заявки, к которой привязывается гость
idTG = идентификатор пользователя TG, используется не для всех
GuestNameAutoNumber = Фамилия, имя участника, номер и марка авто
Type (int) = тип заявки
Role (text) = Роль участника в проекте
Status (int) = состояние рассмотрения участника

Типы

1 = Гость
2 = Автомобиль

Состояния

0 = Не рассмотрен
1 = Подтверждён
2 = Отказ
3 = Автоподтверждение

Настройки

Бот должен иметь следующие настройки, которые вынесены в конфигурационный файл:

- defaultLimit = 20, ограничение количества людей в заявке по-умолчанию
- defaultLimitAuto = NULL, ограничение количества авто в заявке по-умолчанию. Если значение параметра настройки отсутствует или равно NULL, то лимит не применяется.
- stagelsFinal = 9, статус, который определяет финальную стадию рассмотрения заявки. Должна соответствовать одному из доступных значений в перечислении в “Этап рассмотрения заявки” (контролируется администратором).

Критерии приёмки

- Бот запущен и работает в исполняемой области на виртуальной машине
- Код бота залит в Git в соответствующую подпапку

- Объект Администратор добавляется в БД бота после подтверждения суперадмином
- Объект Мероприятие добавляются в БД в таблицу мероприятий
- Объект “Заявка” сохраняется в БД в таблицу с привязкой к мероприятию, для объекта сохраняется признак подтверждения (или отказа) регистрации после подтверждения администратором.
- Все указанные команды обрабатывают в соответствии с описанием
 - Возможна корректировка ТЗ по согласованию с разработчиком при детализации или изменении требования и реализации

Дальнейшее развитие

- Отправка в телеграм билеты
- Сбор участников через телеграм индивидуально - каждый указывает свои данные, включая почту, роль на проекте, имя и прочее.

Прочее

TG @jbShanti

[Ссылка на это ТЗ](#)