**Техническое задание приложения «Планировщик записей клиентов»**

**Описание приложения:**

Приложение разрабатывается для оптимизации и увеличения эффективности ведения записи клиентов на СТО.

*Регистрация*: регистрация проходит на первом запуске приложения, новый пользователь вводит имя, фамилию, название организации. После регистрации, пользователь получает логин и пароль для доступа к организации.

*Авторизация*: авторизация происходит посредством ввода полученных данных для входа.

*Функционал приложения:*

*-* Формировать запись клиентов на определенную дату и время

- Фиксация записи и отслеживание ее статуса

- Доступ к карточкам клиентов, доступ к их автомобилям

- Просмотр записей за определенный период времени (1,2…5 дней и тд)

- Просмотр записи текущего дня

- Просмотр определенной записи (дата, время, причина обращения и тд.)

- Редактирование записи в определенный день

- Оповещение клиентов о предстоящей записи и завершении обслуживания

-

*Описание работы приложения:*

*Новая запись:*

авторизация 🡪 выбор предприятия 🡪 открытие основного меню управления приложения (Записи на сегодня, Создать запись, Записи за определенный период времени, Клиенты и тд.) 🡪 Создать запись 🡪 Выбор даты, указание причины обращения, выбор клиента и авто (по госномеру авто) или создание нового клиента и авто, выбор исполнителя 🡪 Сохранение записи

*Запись на текущий день:*

авторизация 🡪 выбор предприятия 🡪 открытие основного меню управления приложения 🡪 Запись на текущий день 🡪 Вывод всех записей на текущий день или уведомление об отсутствии записей

*Запись за N дней:*

авторизация 🡪 выбор предприятия 🡪 открытие основного меню управления приложения 🡪 Запись за период 🡪 Ввод периода 🡪 Вывод всех записей по дням за N дней

*Просмотр определенной записи:*

авторизация 🡪 выбор предприятия 🡪 открытие основного меню управления приложения 🡪 Выбор пункта календаря 🡪 Выбор дня 🡪 Выбор Записи 🡪 Вывод данных о записи

*Просмотр всех клиентов:*

авторизация 🡪 выбор предприятия 🡪 открытие основного меню управления приложения 🡪 Выбрать пункт «Клиенты»🡪 Вывод всех клиентов

*Уведомление клиента о записи / завершении записи:*

За один час до наступления времени записи клиенту высылается уведомление о предстоящей записи по выбранному каналу связи ( тг, вотс, смс)

**Предварительные ручек (handlers) для реализации функционала приложения:**

**- reg\_handler**

Пользователь заходит в бота тг через свой аккаунт в телеграм – Отправка в базу данных с тг – Создание ключа доступа – Добавление предприятия – Возвращает True/Flase , если True – Вход в в приложение, если False – корректировка данных

**- auth\_handler**

Подумать как реализована фича не выхода из приложения тг продолжительное кол-во времени !!!! Запрос в базу – Возвращает True/False (проверка актуальности ключа)

**- get\_current\_record\_handler (Возвращает текущую запись по времени)**

{

record\_date(15:00): {

car\_brand\_model : “Lada Largus”

record\_reason : “ Троит на холостых”

client\_short\_name : “Владимир М.”

Client\_phone : ‘’89039460606’’

Executor\_short\_name : ‘’Сергей Ф.’’

Record\_time\_over : “16:00”

}

}

**- get\_today\_record\_handler (Возвращает данные о записи на СЕГОДНЯ)**

{

record\_date(15:00): {

car\_brand\_model : “Lada Largus”

record\_reason : “ Троит на холостых”

client\_short\_name : “Владимир М.”

Client\_phone : ‘’89039460606’’

Executor\_short\_name : ‘’Сергей Ф.’’

Record\_time\_over : “16:00”

}

record\_date: …

record\_date: …

}

**Предварительные модели классов для реализации функционала приложения:**

**Модель ЗАПИСЬ (Record)**

**Свойства ЗАПИСЬ (Record):**

- record\_id (уникальный идентификатор записи) type: str

- record\_time\_start (дата и время начала работ) type: datetime

- record\_time\_over (дата и время окончания работ) type: datetime

- record\_reason (причина обращения / цель обращения) type: str

- client\_id (уникальный идентификатор клиента) type: client\_id

- car\_id (уникальный идентификатор автомобиля) type: car\_id

- executor\_id (исполнитель) type: executor\_id

- record\_status (статус записи Подтверждена / В процессе / завершена / не состоялась) type: Enum Str

**Действия с ЗАПИСЬ (Record)**

- создание новой записи

- редактирование существующей записи (дата record\_date, статус record\_status)

- удаление записи

- изменение статуса record\_status (по истечению отведенного времени на работы или самостоятельно в меню управления)

- просмотр записи

**Модель КЛИЕНТ (Customer / Client)**

**Свойства КЛИЕНТ (Customer / Client):**

- client\_id (уникальный идентификатор) type: str

- first\_name (имя клиента) type: str

- last\_name (фамилия клиента) type: str

- phone\_number (номер телефона) type: str

- date\_of\_birth (дата рождения) type: datetime (optional)

- type\_of\_subject (физическое – по умолчанию / юридическое лицо) type: Enum Str

- cars\_list (перечень авто в собственности) type: list[сar\_id]

- notification\_method (способ оповещения клиента о записи, окончании записи, гарантийный номер на работы) type: Enum Str

**Действия с КЛИЕНТ (Customer / Client)**

- создание клиента

- редактирование клиента

- добавление нового авто

**Модель АВТОМОБИЛЬ (Car)**

**Свойства АВТОМОБИЛЬ (Car):**

- car\_id (уникальный идентификатор ГОСНОМЕР) type: str

- car\_brand (Марка Авто) type: str

- car\_model (Модель Авто) type: str

- car\_owner (Владелец авто) type: client\_id

- car\_VIN (VIN номер авто) type: str (optional)

- record\_id (список id записей обслуживания авто) type: list[record\_id] **ПОДУМАТЬ НАДО ЛИ**

**Действия с АВТОМОБИЛЬ (Car)**

- создание нового авто

- редактирование авто

- удаление авто ( у клиента и совсем)

**Модель ИСПОЛНИТЕЛЬ (Executor)**

**Свойства ИСПОЛНИТЕЛЬ (Executor):**

- executor\_id

- executor\_first\_name type:str

- executor\_last\_name type:str

- executor\_record\_list Список записей мастера type list[record\_id]

**Действия с ИСПОЛНИТЕЛЬ (Executor)**

- создание мастера

- редактирование мастера

- удаление мастера

- добавление мастера в запись (Record)