Референсы

* https://glovoapp.com/

Структура:

1. Главная страница

1.1. Если пользователь не авторизован то показывает модальное окно регистраций…

1.1.1 Выбор места ( Кафешек ) или через поиск по названию.

1.1.2 Видит информацию о кафе (не проваливается если кафешка заполнена)

1.1.2.1 Фото

1.1.2.2 Название кафе

1.1.2.3 Адрес

1.1.2.4 Количество

1.1.2.4.1 Если количество больше одного то можно листать по ним.

1.1.2.5 Виды кухни (Опционально)

1.1.3 Выбор количество людей.

1.1.3.1 Показывать свободные столы которые есть.

1.1.4 Выбор “Буду через”

1.1.4.1 Опций [10 минут, 15 минут, 20 минут, 30 минут, час, 2 часа, 4 часа, или кастомное время и дата]

1.1.5 Приходит смс с кодом подтверждения (код подтверждения действителен 1 минуту)

1.1.5.1 Если правильный код то модальное окно об успешном броне.

1.5.1.1.2 Добавление брони в список броней.

1.5.1.1.3 Переход на главную страницу.

1.1.5.2 Если неправильный код то выходит сообщение об ошибке

1.2 Если пользователь авторизован то он остается на главной странице.

1.2.1 Кнопка “мои брони”

1.2.1.1 Если есть бронь то показывает их список

1.2.1.2 Если у юзера много броней в одной кафешке то впихивает их в один селект.

1.2.1.1.1 Юзер проваливается по конкретной брони

1.2.1.1.1.1 Может изменить время брони

1.2.1.1.1.2 Отменить бронь

1.2.2 Кнопка мой аккаунт

1.2.2.1 Видно информация о пользователе (только username, first\_name, last\_name).

1.2.2.1.1 Во вкладке информация он может видеть свой номер телефона.

1.2.2.1.2 Изменить свой юзернейм, ФИО

1.2.2.1.3 Изменить телефон через смс

1.2.2.1.4 Удалить аккаунт (Тогда удаляться все его данные, и брони);

Данные:

1. кафе:

cafe = {

id: string,

photo: string,

name: string,

address: string,

count: number,

cuisine\_types: arrayOf({id: string, name: string}),

cafes\_list: arrayOf(cafe),

phone\_number: string,

full: boolean,

places: places,

}

Список кафешек:

cafes\_list = {

id: string,

list: arrayOf({cafeId: string})

}

1. Данные пользователя:

user = {

id: string,

first\_name: string,

last\_name: string,

phone\_number: string,

username: string,

reserve\_list: reserve\_list\_id,

}

1. Опций времени arrayOf({id: string, duration: seconds, type: [‘hours’’, ‘minutes’]’})
2. Список броней reserve\_list = {

id: string,

userId: string,

}

1. Бронь

reserve = {

id: string,

time: string,

cafe: cafeId,

reserve\_list\_id: string,

places: arrayOf(place),

status: oneOf([‘active’, ‘passed’])

}

1. Место

place = {

id: string,

min: number,

max: number,

isReserved: boolean,

cafeId: string

}

HTTP Запросы.

GET список кафешек: \cafes

GET кафе: \cafes\{:cafeId}

GET все места в одном кафе: /cafes/{:cafeid}/places/

GET список броней: \reserves

POST новая бронь или GET конкретная бронь: \reserves\{reserveId}

PATCH менять инфу о броне (время) \reserves\{reserveId}

DELETE удаление брони \reserves\{reserveId}

GET получить инфу о юзере: user\{username}

PATCH менять инфу о юзере user\{username}

DELETE удаление аккаунта \user\{username} => DELETE удаление брони \reserves\{reserveId}

**Архитектура Базы Данных**

* + - 1. Subject Area Analysis.

В базе данных будут хранится следующие данные: пользователи (имя, фамилия, номер телефона, юзернэйм/никнейм), кафе (название, фото, адрес, телефон, наполненность), виды кухни, кафе с указанием кухни (их может быть несколько), опции времени, брони (привязка пользователя к определенному слоту времени и месту в кафе), места (в определенных кафе с указанием минимального и максимального количества гостей).

* + - 1. Conceptual Design.

Table 1. Users

|  |  |  |
| --- | --- | --- |
| Attribute | Data type | Constraint / Note |
| id | INTEGER | PRIMARY KEY |
| username | VARCHAR(50) | UNIQUE |
| first\_name | VARCHAR(50) | NOT NULL |
| last\_name | VARCHAR(50) | NOT NULL |
| phone\_number | VARCHAR(12) | UNIQUE  max\_length = 12  pattern => +7 111 222 33 44 |

Table 2. Cafes

|  |  |  |
| --- | --- | --- |
| Attribute | Data type | Constraint / Note |
| id | INTEGER | PRIMARY KEY |
| name | VARCHAR(50) | NOT NULL |
| photo | VARCHAR / ImageField | NOT NULL |
| address | VARCHAR(200) | NOT NULL |

Table 3. Cafes\_Phone\_Numbers

|  |  |  |
| --- | --- | --- |
| Attribute | Data type | Constraint / Note |
| id | INTEGER | PRIMARY KEY |
| phone\_number | VARCHAR(12) | UNIQUE  max\_length = 12  pattern => +7 111 222 33 44 |
| café\_id | INTEGER | FOREIGN KEY cafés(id) |

Table 3. Cuisines

|  |  |  |
| --- | --- | --- |
| Attribute | Data type | Constraint / Note |
| id | INTEGER | PRIMARY KEY |
| name | VARCHAR(50) | UNIQUE |

Table 4. Café\_Cuisines

|  |  |  |
| --- | --- | --- |
| Attribute | Data type | Constraint / Note |
| id | INTEGER | PRIMARY KEY |
| café\_id | INTEGER | FOREIGN KEY cafes(id) |
| cuisines\_id | INTEGER | FOREIGN KEY cuisines(id) |

Table 5. Reserve

|  |  |  |
| --- | --- | --- |
| Attribute | Data type | Constraint / Note |
| id | INTEGER | PRIMARY KEY |
| user\_id | INTEGER | FOREIGN KEY users(id) |
| place\_id | INTEGER | FOREIGN KEY places(id) |
| start\_time | DATETIME | NOT NULL |
| duration | INTEGER | NOT NULL |

Table 6. Place

|  |  |  |
| --- | --- | --- |
| Attribute | Data type | Constraint / Note |
| id | INTEGER | PRIMARY KEY |
| café\_id | INTEGER | FOREIGN KEY cafes(id) |
| min\_guest\_number | INTEGER | NOT NULL |
| max\_guest\_number | INTEGER | NOT NULL |