**КФС:**

**Приложение должно состоять из 3 частей:**

**1 часть для самого клиента:**

В ней должно быть меню выбора блюд. В меню должно содержаться название блюда или комплект еды, цена блюд, комплектов и т.д., калорийность каждого блюда, а так же корзина.

Клиент может добавлять разные блюда в корзину в неограниченном количестве.

После выбора товара, клиент должен видеть общую сумму взятых блюд.

Он делает платёж, ему показывается его место в очереди, данные о покупке клиента отправляются повару.

Так же есть приложение на телефон. В нем показывается функционал программы, а так же состояние о заказе клиента (готов, в работе, в очереди).

В приложении должна быть функция регистрации и авторизации клиентов. Когда клиент в первый раз заходит в приложение, ему будет выдана информация о самом приложении и будет предложено зарегистрироваться.

Незарегестрированным пользователям будут недоступны функции приложения и состояние заказа. У каждого зарегистрированного пользователя будет свой ID, который он будет вводить при оплате заказа.

Будут начисляться баллы, которые нужны для скидок.

Так же в приложении будет доставка.

**2 часть для повара:**

Приложение будет запущено на планшете или небольшом сенсорном мониторе. Интерфейс программы должен быть удобен для использования именно сенсорного ввода. В ходе своей работы повару нужно нажимать необходимые кнопки в приложении пальцем, поэтому интерфейс должен быть рассчитан на ситуации, когда он в спешке может не попасть по кнопке. Так же будет осуществлён голосовой ввод.

При команде "Заказ" приложение открывает первый заказ в очереди и в заказе на приложение клиента будет написано "в работе". Интерфейс программы должен разделять визуально экран на две части: слева он видит список блюд в очереди; справа состав блюда, которое он выбрал для приготовления, а так же способы приготовления ингрeдиентов. Выбор блюда происходит нажатием на блюдо в списке слева. Как только он на него нажал, блюдо приобретает статус «В работе», из списка слева он удаляется, а с правой стороны он видит его состав. Так же справа должно быть две кнопки: «Отменить» (красного цвета), «Готово» (зеленого цвета). Если нажать на «Отменить», то оно возвращается в список очереди. Если нажать на «Готово», то блюдо приобретает статус «Готов к выдачи» и правая сторона програмы становится пустой, пока повар не выберет следующее блюдо. Ещё одна кнопка будет отвечать за выдачу блюд клиентам. Если на неё нажать, то все блюда со статусом «Готов к выдачи» приобретают статус «Выполнен», а из сумма прибавляется к сумме выручке сети, которую необходимо отображать в правом верхнем углу.

**3 часть для официанта:**

В ней показывается Заказ Клиента(Какое блюдо он заказал, его готовность, столик клиента и общая стоимость заказа).

После того как повар нажал на кнопку "Выдача" и блюдо получает статус "Готов к выдаче", на экран официанта сообщение, что в заказе n готово блюдо a, это последнее блюдо, или же в заказе есть еще блюда в очереди.

\создание БД

CREATE TABLE [Ingredient] (

[Name] varchar(60) NOT NULL,

Calories integer NOT NULL,

CONSTRAINT [PK\_INGREDIENT] PRIMARY KEY CLUSTERED

(

[Name] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [Dish] (

[Name] varchar(100) NOT NULL,

Cost money NOT NULL,

[Description] varchar(255) NOT NULL,

CONSTRAINT [PK\_DISH] PRIMARY KEY CLUSTERED

(

[Name] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [Unit] (

[Name] varchar(12) NOT NULL,

CONSTRAINT [PK\_UNIT] PRIMARY KEY CLUSTERED

(

[Name] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [DishSet] (

[Name] varchar(255) NOT NULL,

Dish varchar(100) NOT NULL,

[Count] integer NOT NULL,

CONSTRAINT [PK\_DISHSET] PRIMARY KEY CLUSTERED

(

[Name] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [DishCompound] (

DishName varchar(100) NOT NULL,

IngName varchar(60) NOT NULL,

Quantity int NOT NULL,

Unit varchar(12) NOT NULL,

CONSTRAINT [PK\_DISHCOMPOUND] PRIMARY KEY CLUSTERED

(

[DishName] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [Order] (

NumberOrder integer NOT NULL,

[Status] varchar(20) NOT NULL,

CONSTRAINT [PK\_ORDER] PRIMARY KEY CLUSTERED

(

[NumberOrder] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [DishStatus] (

[Status] varchar(20) NOT NULL,

CONSTRAINT [PK\_DISHSTATUS] PRIMARY KEY CLUSTERED

(

[Status] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [OrderCompound] (

OrderNumber integer NOT NULL,

Dish varchar(100) NOT NULL,

Count integer not null,

CONSTRAINT [PK\_ORDERCOMPOUND] PRIMARY KEY CLUSTERED

(

[OrderNumber] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [User] (

[TelephonNum] varchar(12) NOT NULL,

[Password] varchar(10) NOT NULL,

UserID varchar(11) NOT NULL,

Points integer NOT NULL,

CONSTRAINT [PK\_USER] PRIMARY KEY CLUSTERED

(

[UserID] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [MobileOrder] (

NumberOrder integer NOT NULL,

UserID varchar(11) NOT NULL,

[Address] integer NOT NULL,

CONSTRAINT [PK\_MOBILEORDER] PRIMARY KEY CLUSTERED

(

[NumberOrder] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

ALTER TABLE [DishSet] WITH CHECK ADD CONSTRAINT [DishSet\_fk0] FOREIGN KEY ([Dish]) REFERENCES [Dish]([Name])

ON UPDATE CASCADE

GO

ALTER TABLE [DishSet] CHECK CONSTRAINT [DishSet\_fk0]

GO

ALTER TABLE [DishCompound] WITH CHECK ADD CONSTRAINT [DishCompound\_fk0] FOREIGN KEY ([DishName]) REFERENCES [Dish]([Name])

ON UPDATE CASCADE

GO

ALTER TABLE [DishCompound] CHECK CONSTRAINT [DishCompound\_fk0]

GO

ALTER TABLE [DishCompound] WITH CHECK ADD CONSTRAINT [DishCompound\_fk1] FOREIGN KEY ([IngName]) REFERENCES [Ingredient]([Name])

ON UPDATE CASCADE

GO

ALTER TABLE [DishCompound] CHECK CONSTRAINT [DishCompound\_fk1]

GO

ALTER TABLE [DishCompound] WITH CHECK ADD CONSTRAINT [DishCompound\_fk2] FOREIGN KEY ([Unit]) REFERENCES [Unit]([Name])

ON UPDATE CASCADE

GO

ALTER TABLE [DishCompound] CHECK CONSTRAINT [DishCompound\_fk2]

GO

ALTER TABLE [Order] WITH CHECK ADD CONSTRAINT [Order\_fk0] FOREIGN KEY ([Status]) REFERENCES [DishStatus]([Status])

ON UPDATE CASCADE

GO

ALTER TABLE [Order] CHECK CONSTRAINT [Order\_fk0]

GO

ALTER TABLE [OrderCompound] WITH CHECK ADD CONSTRAINT [OrderCompound\_fk0] FOREIGN KEY ([OrderNumber]) REFERENCES [Order]([NumberOrder])

ON UPDATE CASCADE

GO

ALTER TABLE [OrderCompound] CHECK CONSTRAINT [OrderCompound\_fk0]

GO

ALTER TABLE [OrderCompound] WITH CHECK ADD CONSTRAINT [OrderCompound\_fk1] FOREIGN KEY ([Dish]) REFERENCES [Dish]([Name])

ON UPDATE CASCADE

GO

ALTER TABLE [OrderCompound] CHECK CONSTRAINT [OrderCompound\_fk1]

GO

ALTER TABLE [MobileOrder] WITH CHECK ADD CONSTRAINT [MobileOrder\_fk0] FOREIGN KEY ([NumberOrder]) REFERENCES [Order]([NumberOrder])

ON UPDATE CASCADE

GO

ALTER TABLE [MobileOrder] CHECK CONSTRAINT [MobileOrder\_fk0]

GO

ALTER TABLE [MobileOrder] WITH CHECK ADD CONSTRAINT [MobileOrder\_fk1] FOREIGN KEY ([UserID]) REFERENCES [User]([UserID])

ON UPDATE CASCADE

GO

ALTER TABLE [MobileOrder] CHECK CONSTRAINT [MobileOrder\_fk1]

GO