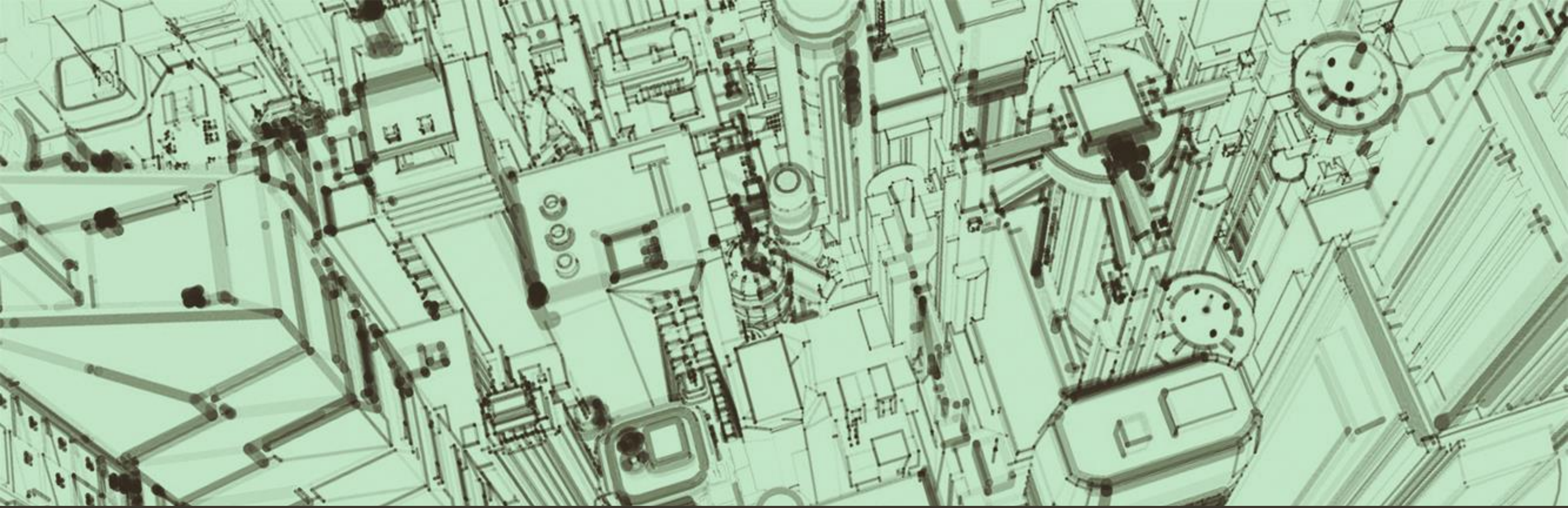


E-KART

An E-Commercial-Business application

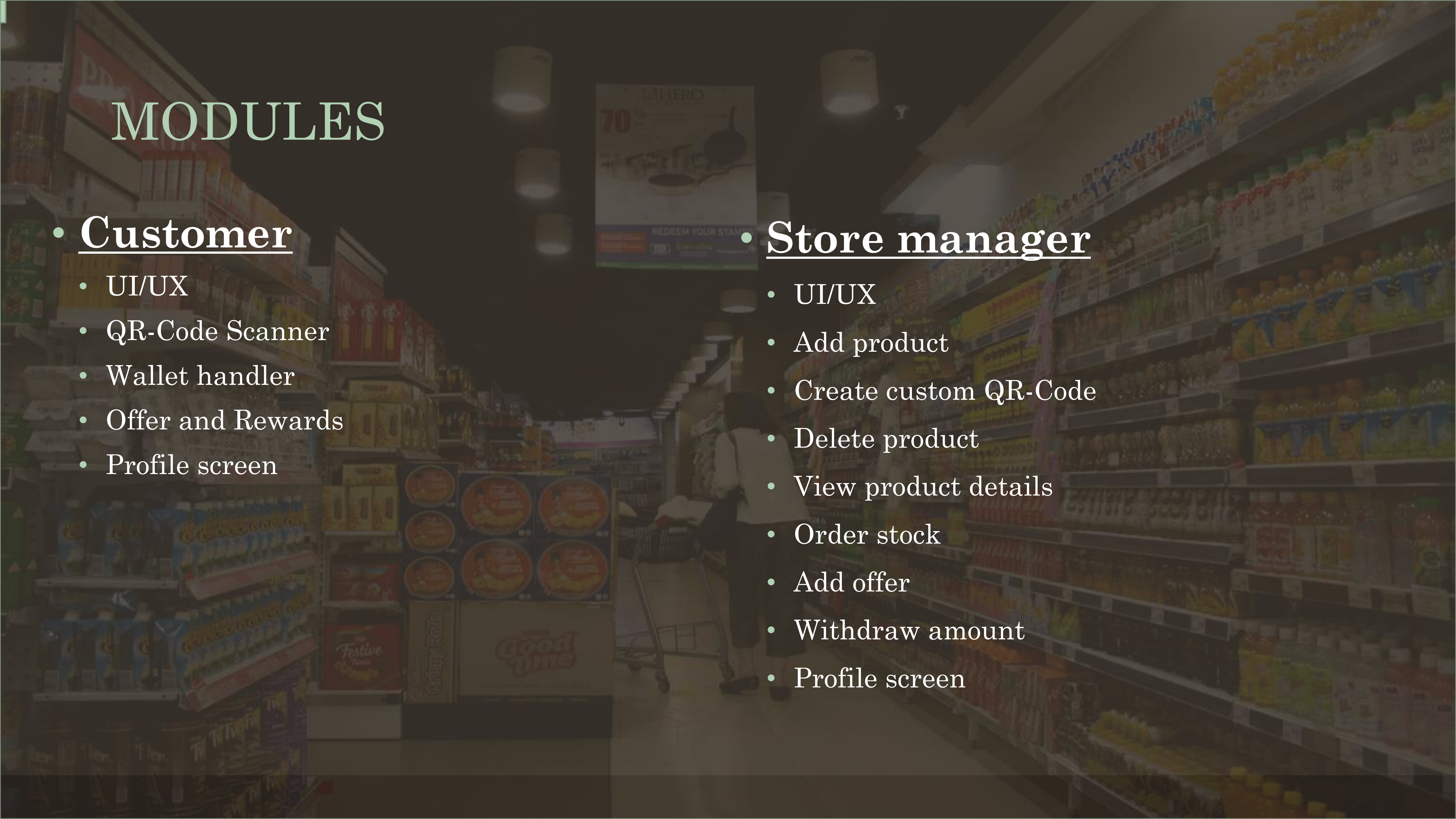
INTRODUCTION

- The main issue while shopping in a super-market or a hypermarket is the long time on the billing counter.
- We introduce this app for reduce the billing time by billing the product by customer it self and payment through the same app.

```

#-----------------------------------

class E_Kart(MDApp):

    def build(self):
        self.theme_cls.theme_style = "D
        self.theme_cls.accent_palette =
        return Builder.load_file("Logi
    # --importing the design and sett

    def Text_Capital(self, instance)
        # -- text capitalize functio
        if instance.text:
            instance.text = instanc

#--------------------------------


if __name__ == "__main__":
    # -- Starting the app --
    E_Kart().run()

```

ABSTRACT

- The E-CART is not totally a e-commerce app.
- It is the combination of offline and online store which customer can choose the product, check the quality, verifying if it working or not, after the customer satisfied he can pay Through the app.
- This app also avoid the waiting time for billing to make a big queue and other delays
- As the store management side that helps to minimize the employees by removing the cashier, butcher and reduce the effort of the store manager



MODULES

- **Customer**
  - UI/UX
  - QR-Code Scanner
  - Wallet handler
  - Offer and Rewards
  - Profile screen

- **Store manager**
  - UI/UX
  - Add product
  - Create custom QR-Code
  - Delete product
  - View product details
  - Order stock
  - Add offer
  - Withdraw amount
  - Profile screen

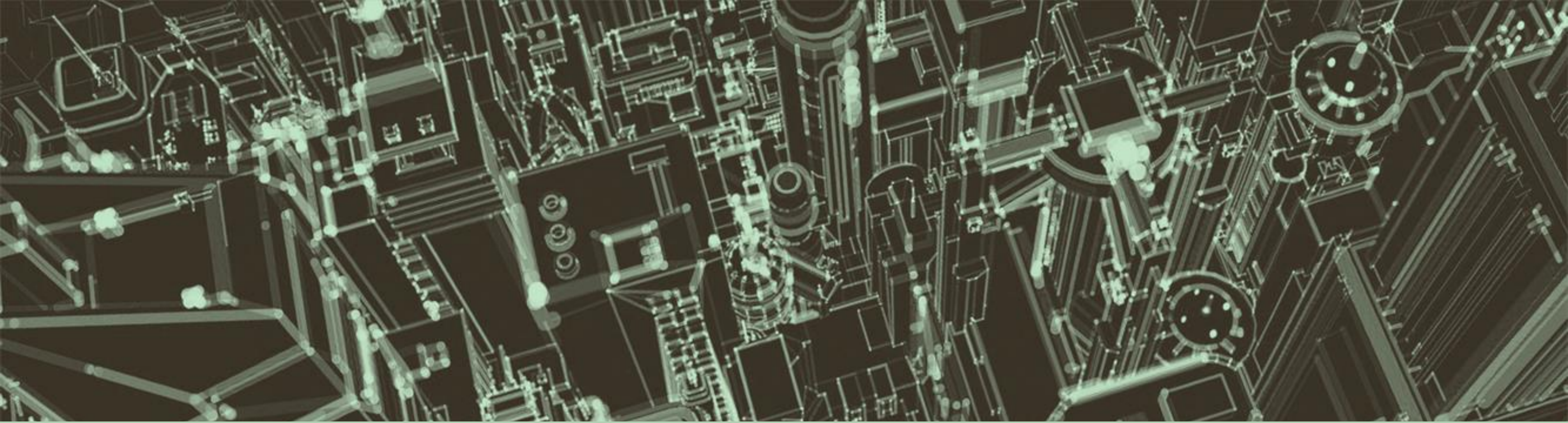
PACKAGES USED FOR THIS APPLICATION

- PYTHON
- PYTHON-KIVY
- PYTHON-KIVYMD

- RANDOM
- KIVY_GARDEN
- BUILDOZER
- PYINSTALLER

PYTHON-KIVY/ KIVYMD

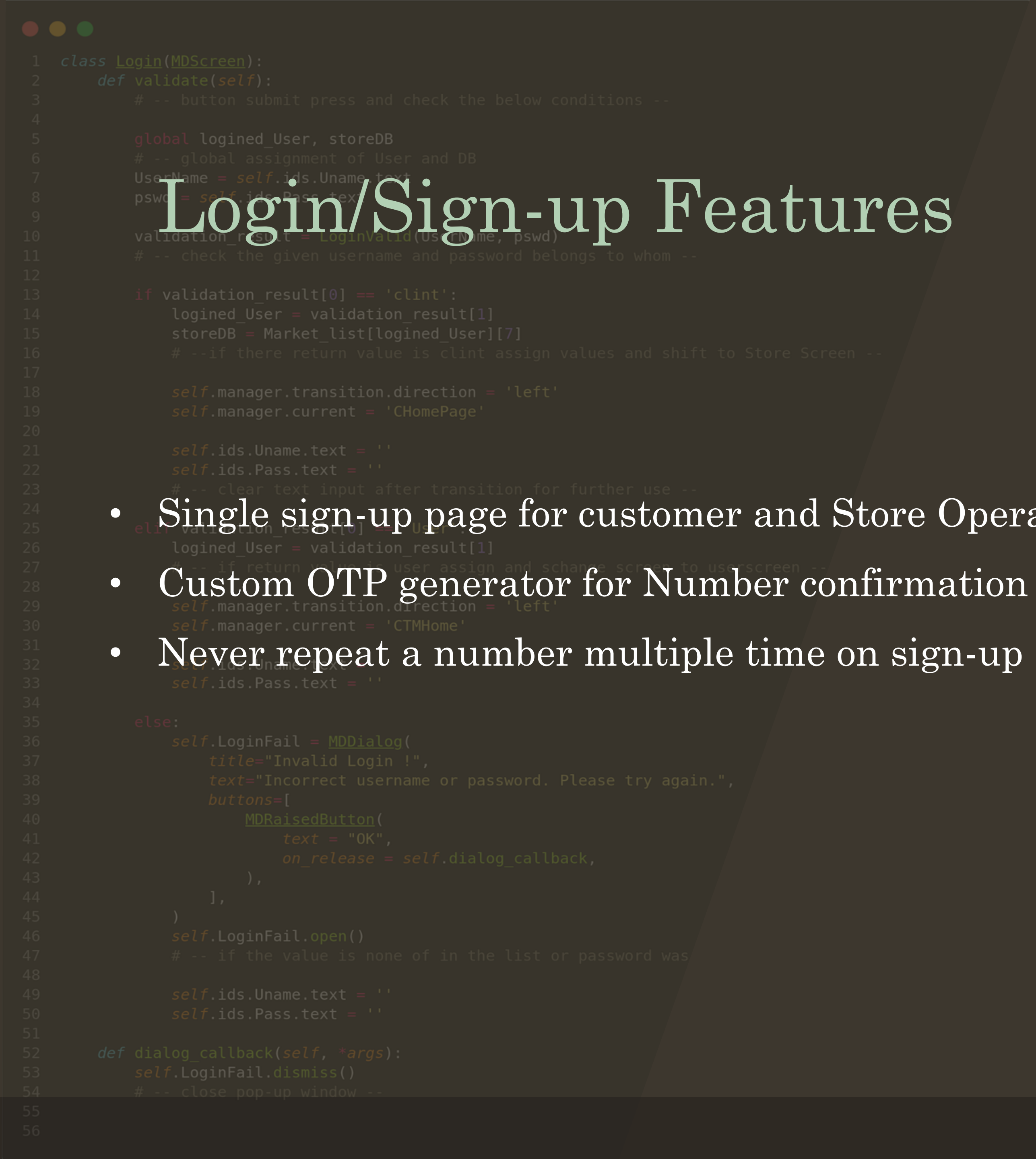
- Kivy is a Python library for developing multitouch applications.
- It’s cross-platform (Linux/OS X/Windows/Android/iOS) and released under the MIT license.
- It is particularly good for applications that require multi-touch, gestures, and other modern touch features.
- KivyMD is an extension of the Kivy framework.
- It is a collection of Material Design compliant widgets for use with Kivy.
- The goal of the project is to approximate Google’s Material Design spec as close as possible without sacrificing ease of use.



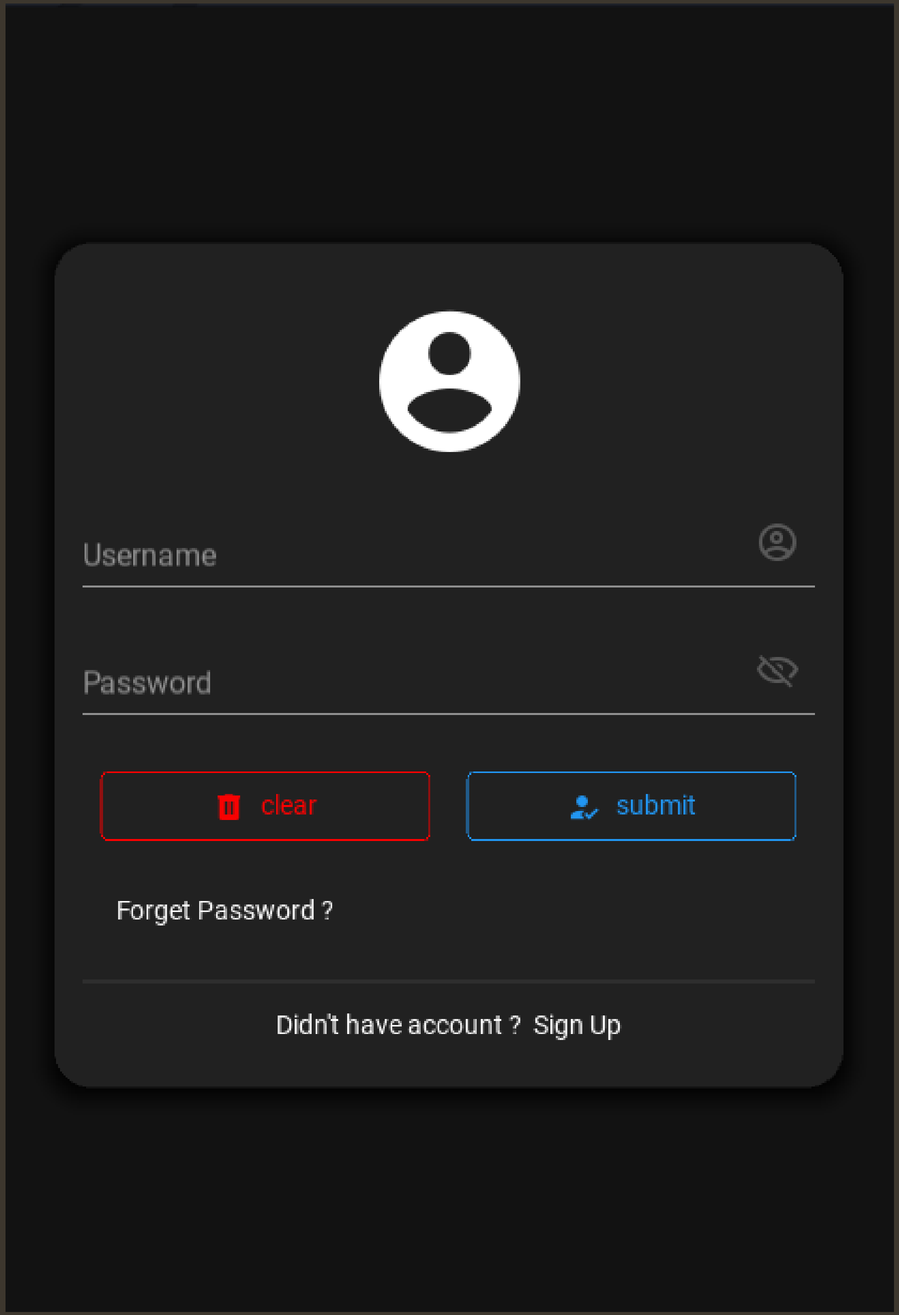
LOGIN/SIGN-UP AND RESET PASSWORD

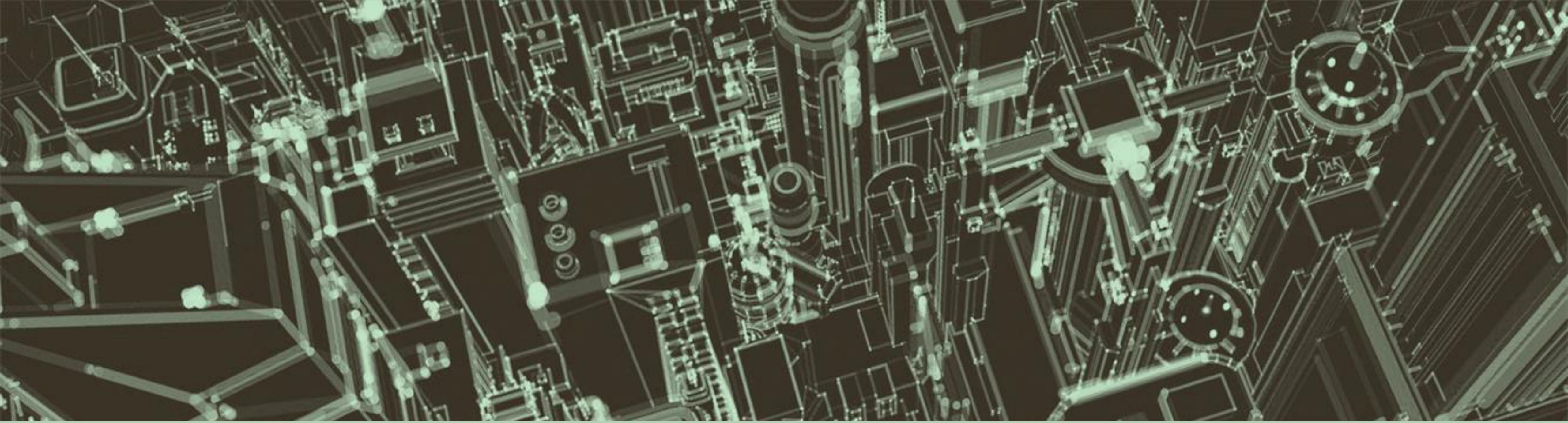
Login/Sign-up Features

- Single sign-up page for customer and Store Operator
- Custom OTP generator for Number confirmation
- Never repeat a number multiple time on sign-up

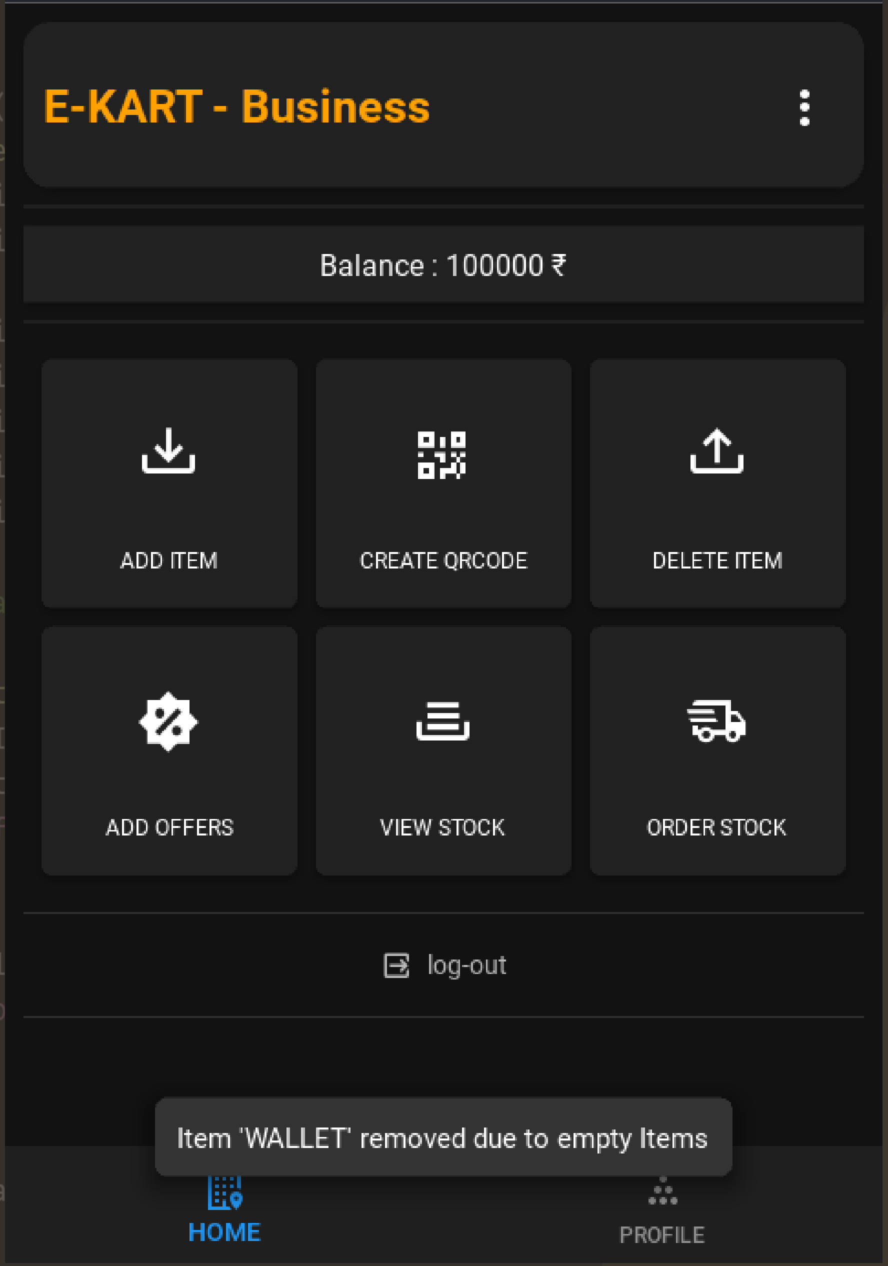


```python
class Login(MDScreen):
    def validate(self):
        # -- button submit press and check the below conditions --

        global logined_User, storeDB
        # -- global assignment of User and DB
        UserName = self.ids.Uname.text
        pswd = self.ids.Pass.text

        validation_result = LoginValid(UserName, pswd)
        # -- check the given username and password belongs to whom --

        if validation_result[0] == 'clint':
            logined_User = validation_result[1]
            storeDB = Market_list[logined_User][7]
            # --if there return value is clint assign values and shift to Store Screen --

            self.manager.transition.direction = 'left'
            self.manager.current = 'CHomePage'

            self.ids.Uname.text = ''
            self.ids.Pass.text = ''
            # -- clear text input after transition for further use --

        elif validation_result[0] == 'User':
            logined_User = validation_result[1]
            # -- if return value is user assign and schange screen to userscreen --

            self.manager.transition.direction = 'left'
            self.manager.current = 'CTMHome'

            self.ids.Uname.text = ''
            self.ids.Pass.text = ''

        else:
            self.LoginFail = MDDialog(
                title="Invalid Login !",
                text="Incorrect username or password. Please try again.",
                buttons=[
                    MDRaisedButton(
                        text = "OK",
                        on_release = self.dialog_callback,
                    ),
                ],
            )
            self.LoginFail.open()
            # -- if the value is none of in the list or password was

            self.ids.Uname.text = ''
            self.ids.Pass.text = ''

    def dialog_callback(self, *args):
        self.LoginFail.dismiss()
        # -- close pop-up window --
```

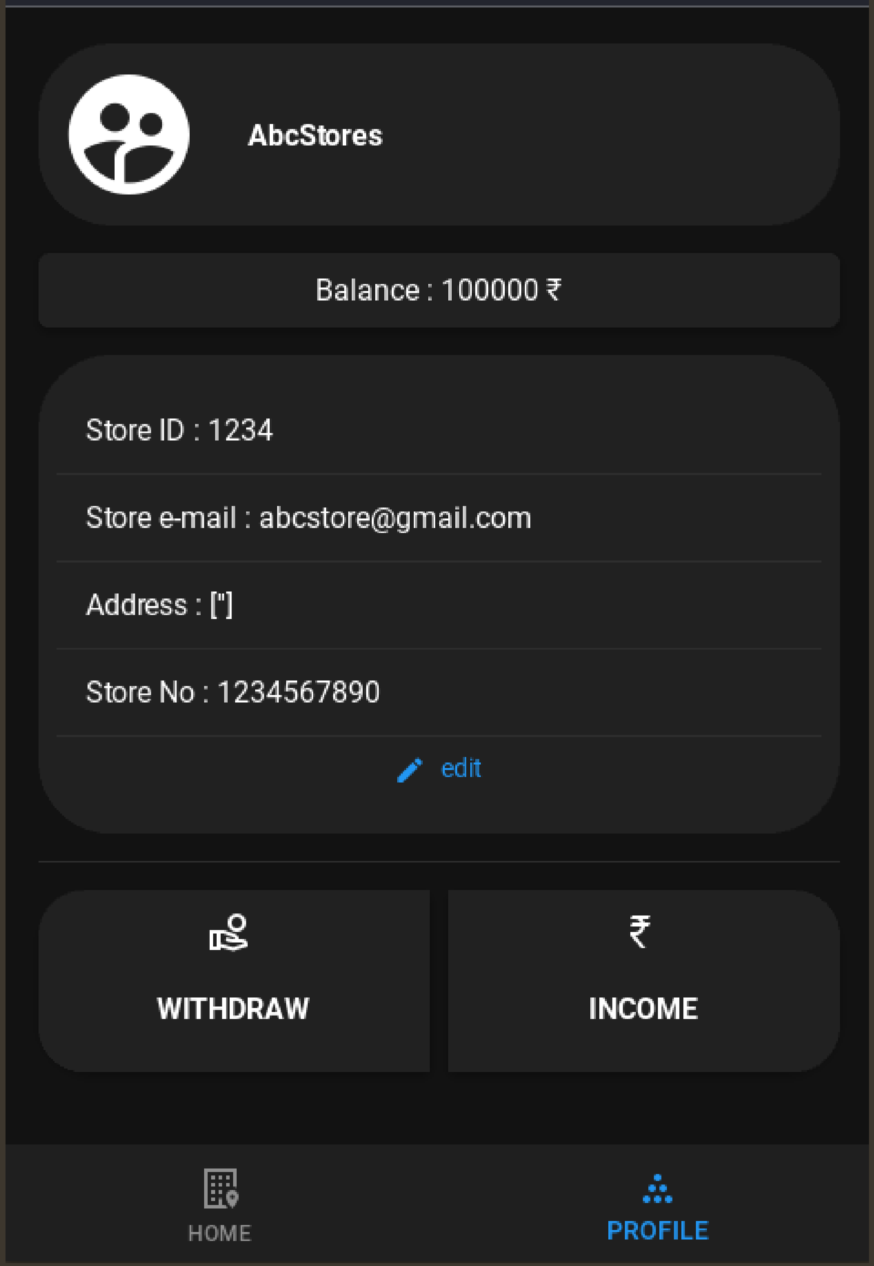




STORE



Item 'WALLET' removed due to empty Items



Balance : 100000 ₹

Store ID : 1234

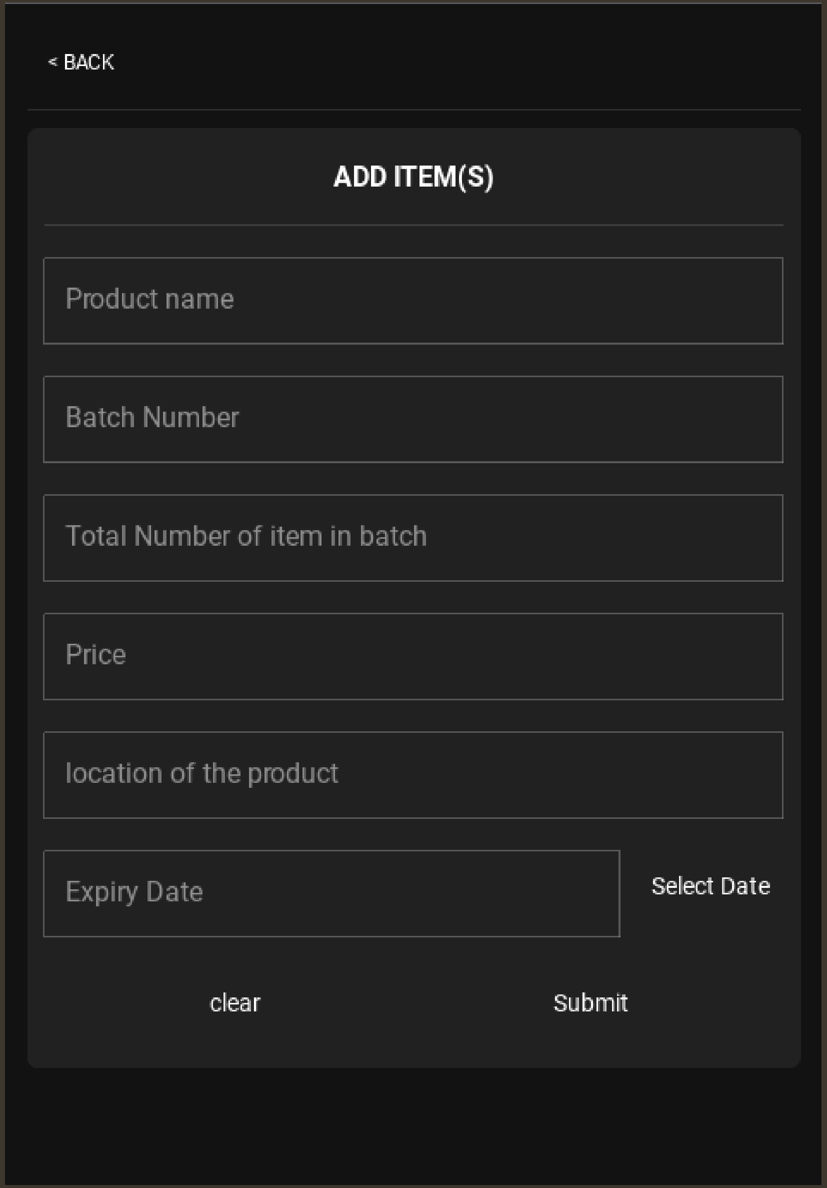
Store e-mail : abcstore@gmail.com

Address : ['']

Store No : 1234567890

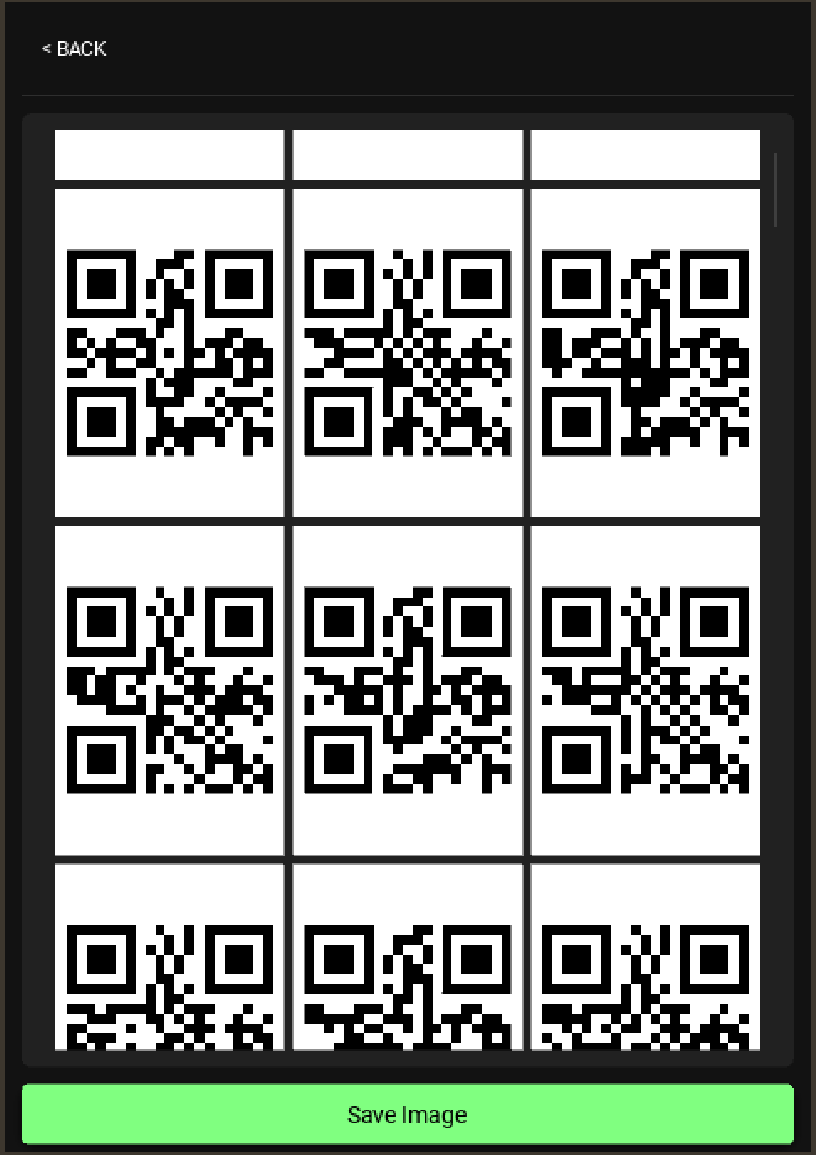
ADDING ITEM TO STORE DB

- A store can add item to their **RACK** (DB)
- Basic details like Product name, Batch No., Total No of Item, Price, Rack Location, EXP Date
- The location can’t share to multiple product
- Each have separate Item ID will assign after Adding item



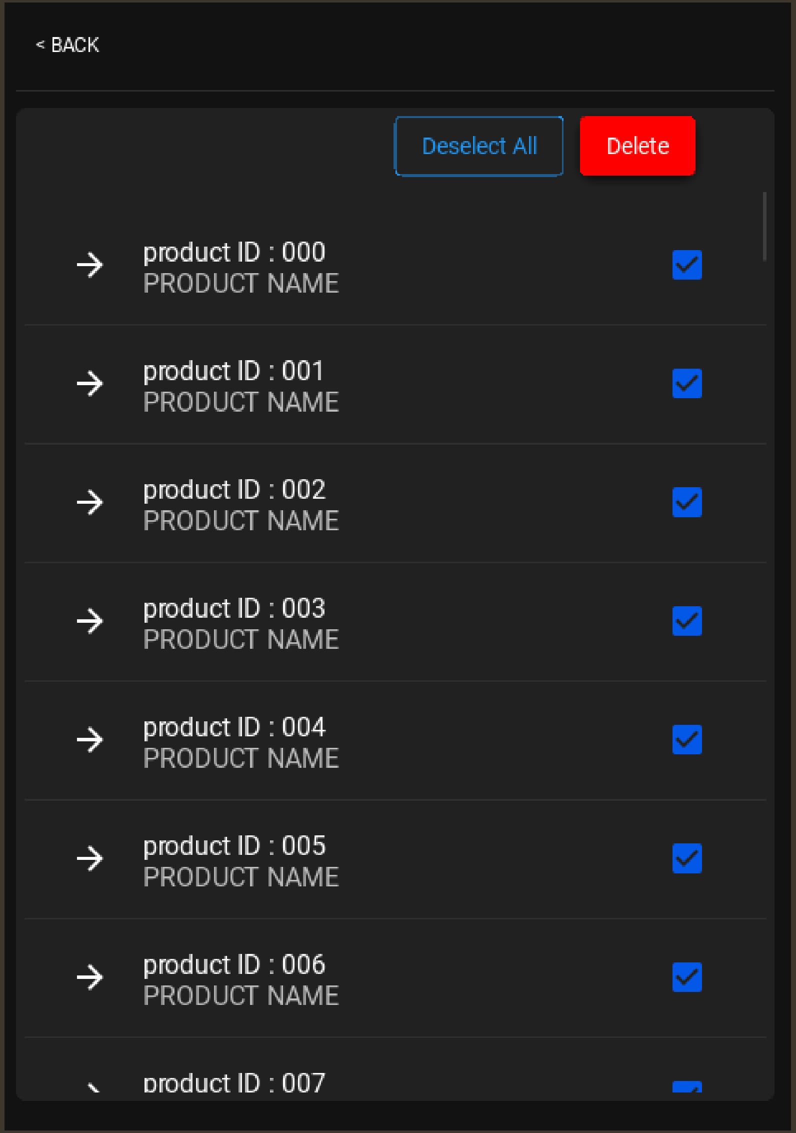
CREATE QR-CODE

- The custom generated code contain the essential details for identify the item and its other details
- Each Qr-code is unique it self
- Store manager can save and print-out for paste on their product
- Each product and item have its own unique ID



DELETE ITEM FROM RACK

- If the product was expired or some unexpected error occur we must delete it from RACK
- We also added delete product not only individually but also delete entire items from that RACK



< BACK

Deselect All

Delete

→ product ID : 000 ☑
PRODUCT NAME

→ product ID : 001 ☑
PRODUCT NAME

→ product ID : 002 ☑
PRODUCT NAME

→ product ID : 003 ☑
PRODUCT NAME

→ product ID : 004 ☑
PRODUCT NAME

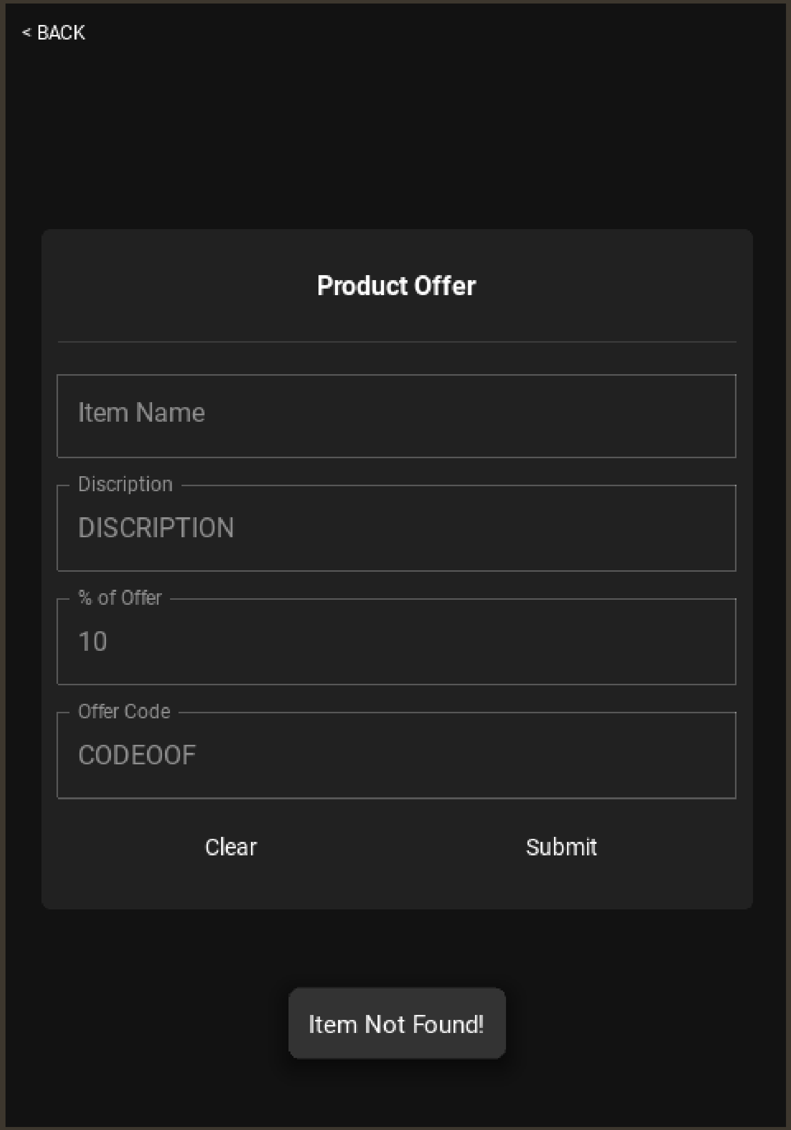
→ product ID : 005 ☑
PRODUCT NAME

→ product ID : 006 ☑
PRODUCT NAME

product ID : 007

ADDING OFFER TO A PRODUCT

- Some time a store need to add offers to their product or entire store for promotion or other reasons
- We can add offer to each product or store by an offer code and it deduct by percent value
- The user can grab that offer while purchasing



Product Offer

Item Name

Discription: DISCRIPTION

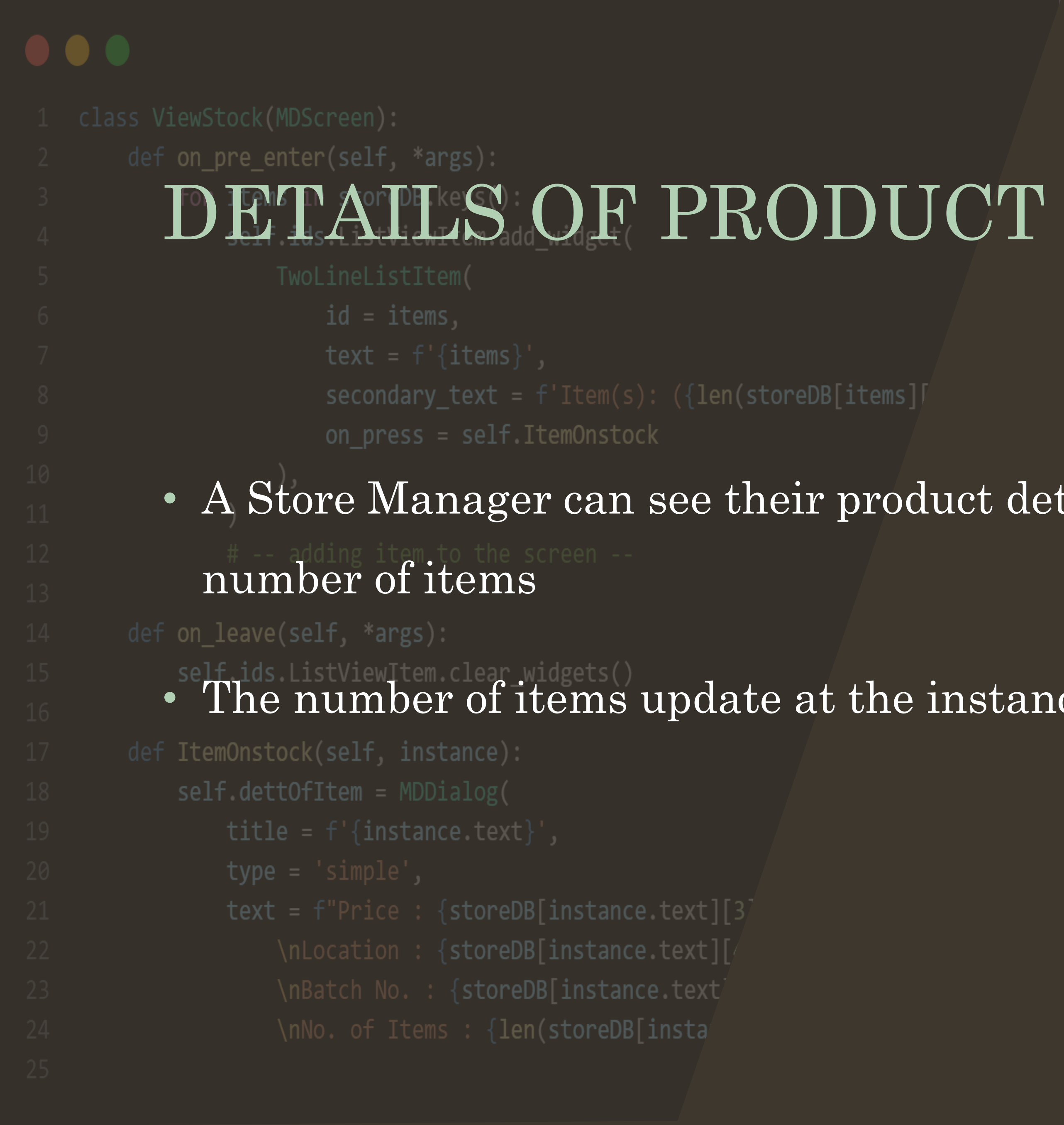
% of Offer: 10

Offer Code: CODEOOF

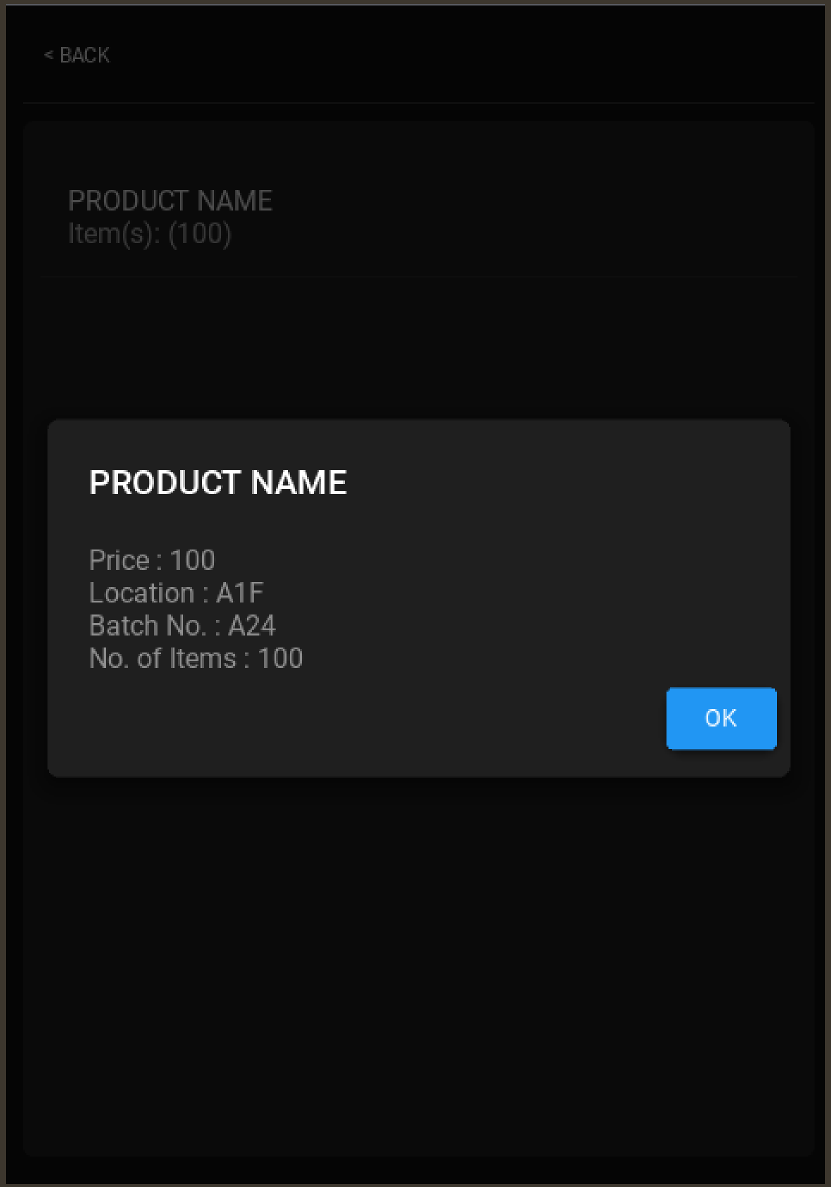
Item Not Found!

DETAILS OF PRODUCT

- A Store Manager can see their product details, number of items
- The number of items update at the instance



```
class ViewStock(MDScreen):
    def on_pre_enter(self, *args):
        for items in storeDB.keys():
            self.ids.ListViewItem.add_widget(
                TwoLineListItem(
                    id = items,
                    text = f'{items}',
                    secondary_text = f'Item(s): ({len(storeDB[items][
                    on_press = self.ItemOnstock
                ),
            )
            # -- adding item to the screen --

    def on_leave(self, *args):
        self.ids.ListViewItem.clear_widgets()

    def ItemOnstock(self, instance):
        self.dettOfItem = MDDialog(
            title = f'{instance.text}',
            type = 'simple',
            text = f"Price : {storeDB[instance.text][3
                \nLocation : {storeDB[instance.text][
                \nBatch No. : {storeDB[instance.text
                \nNo. of Items : {len(storeDB[insta
```



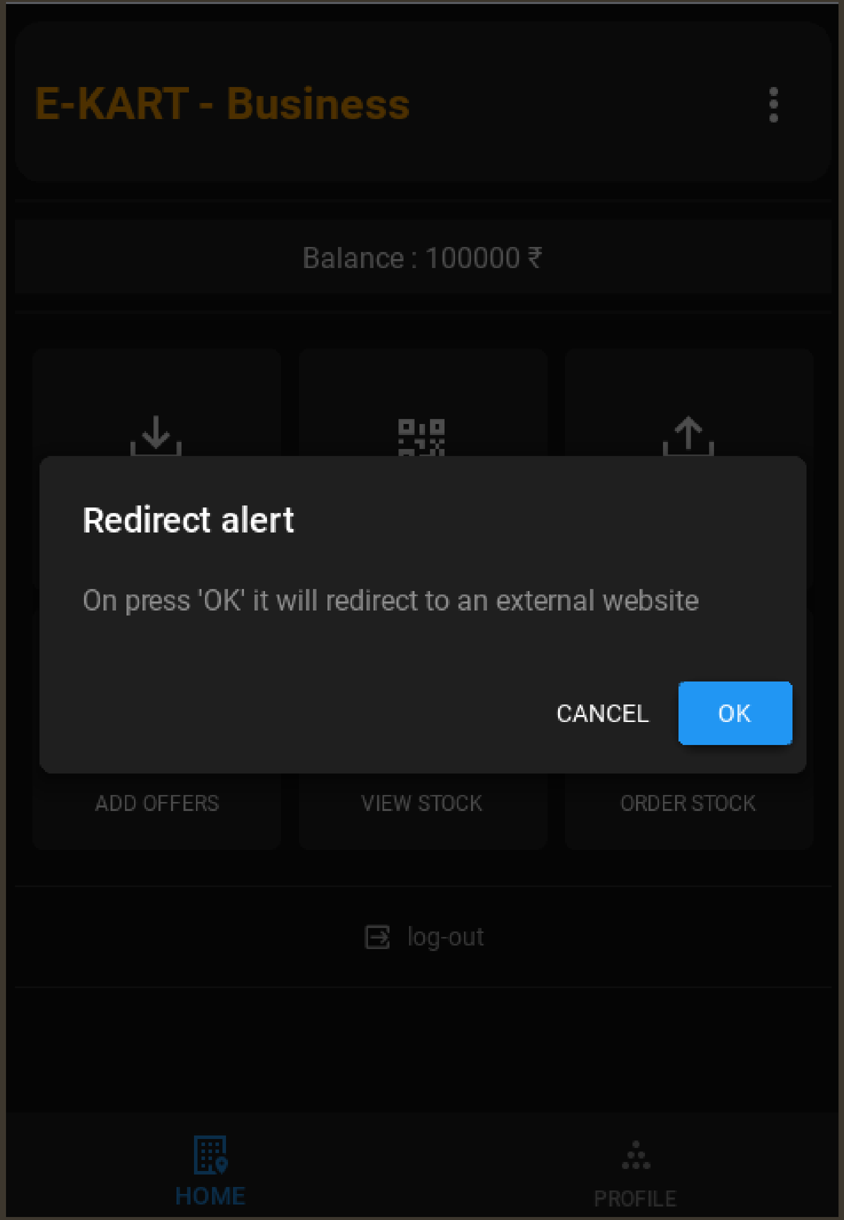
PRODUCT NAME
Item(s): (100)

PRODUCT NAME

Price : 100
Location : A1F
Batch No. : A24
No. of Items : 100

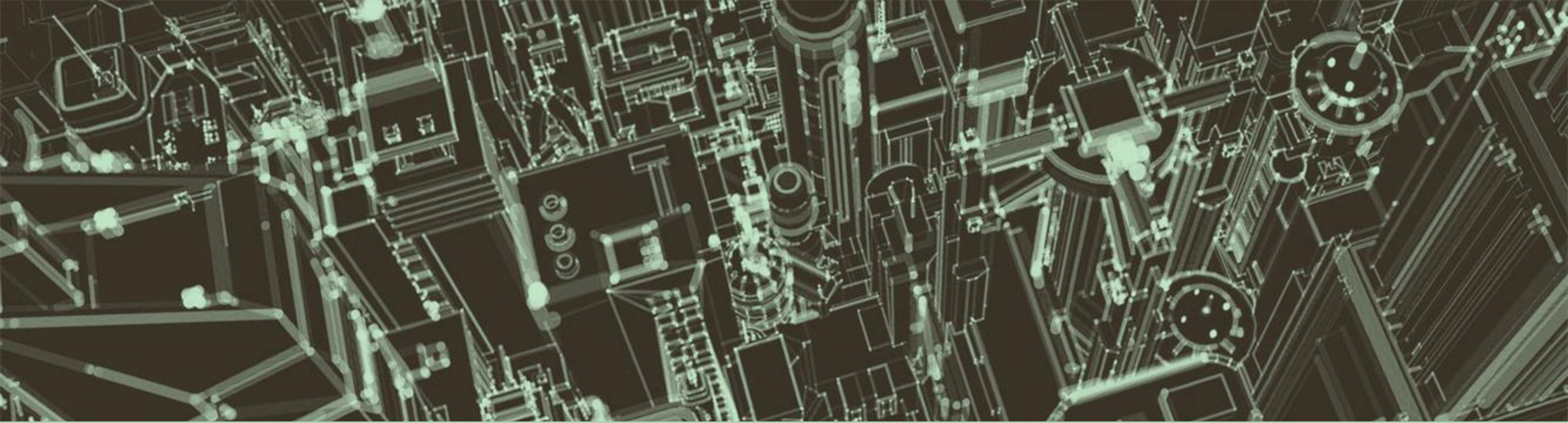
ORDERING STOCKS

- Store can order stocks from Online store or a frequently ordering store by redirecting to their websites
- The amount deduct from the wallet of Store

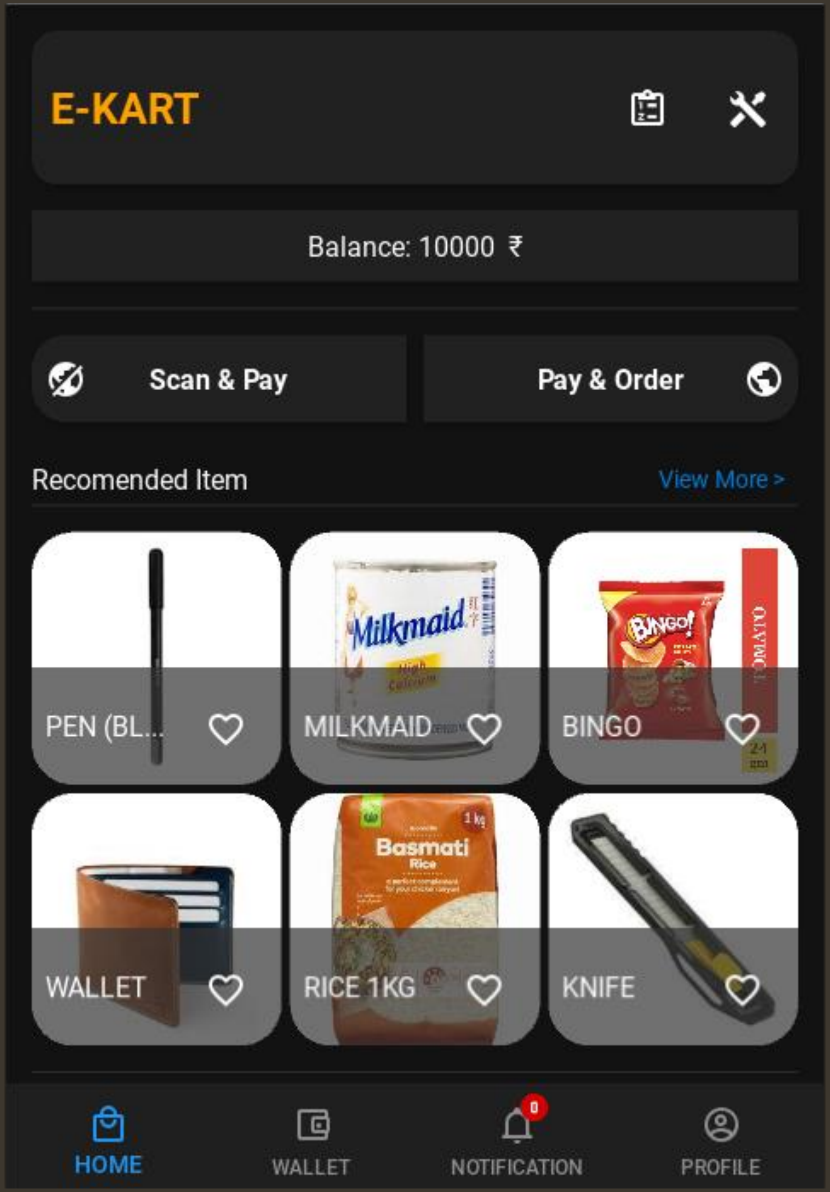


Redirect alert

On press 'OK' it will redirect to an external website



CUSTOMER



E-KART

Balance: 10000 ₹

Scan & Pay

Pay & Order

Recomended Item

View More >

PEN (BL...

MILKMAID

BINGO

WALLET

RICE 1KG

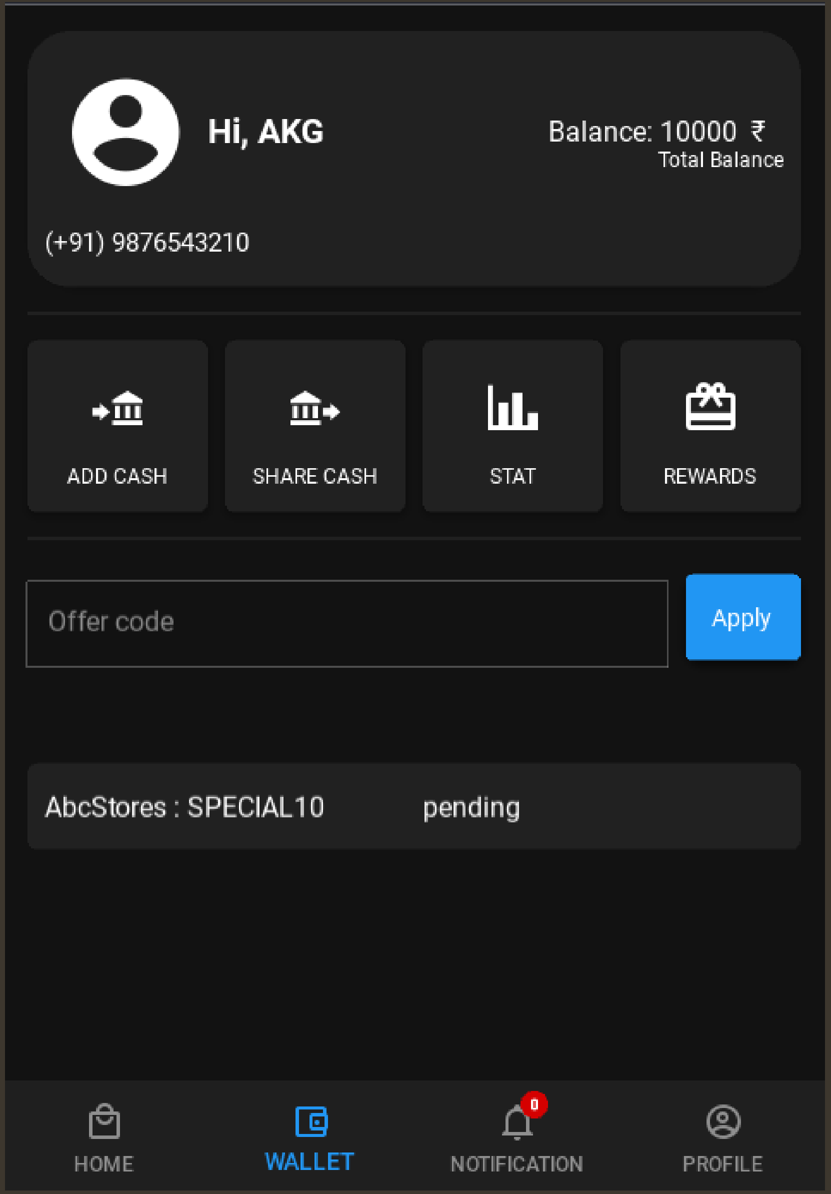
KNIFE

HOME

WALLET

NOTIFICATION

PROFILE



Hi, AKG

Balance: 10000 ₹

Total Balance

(+91) 9876543210

ADD CASH

SHARE CASH

STAT

REWARDS

Offer code

Apply

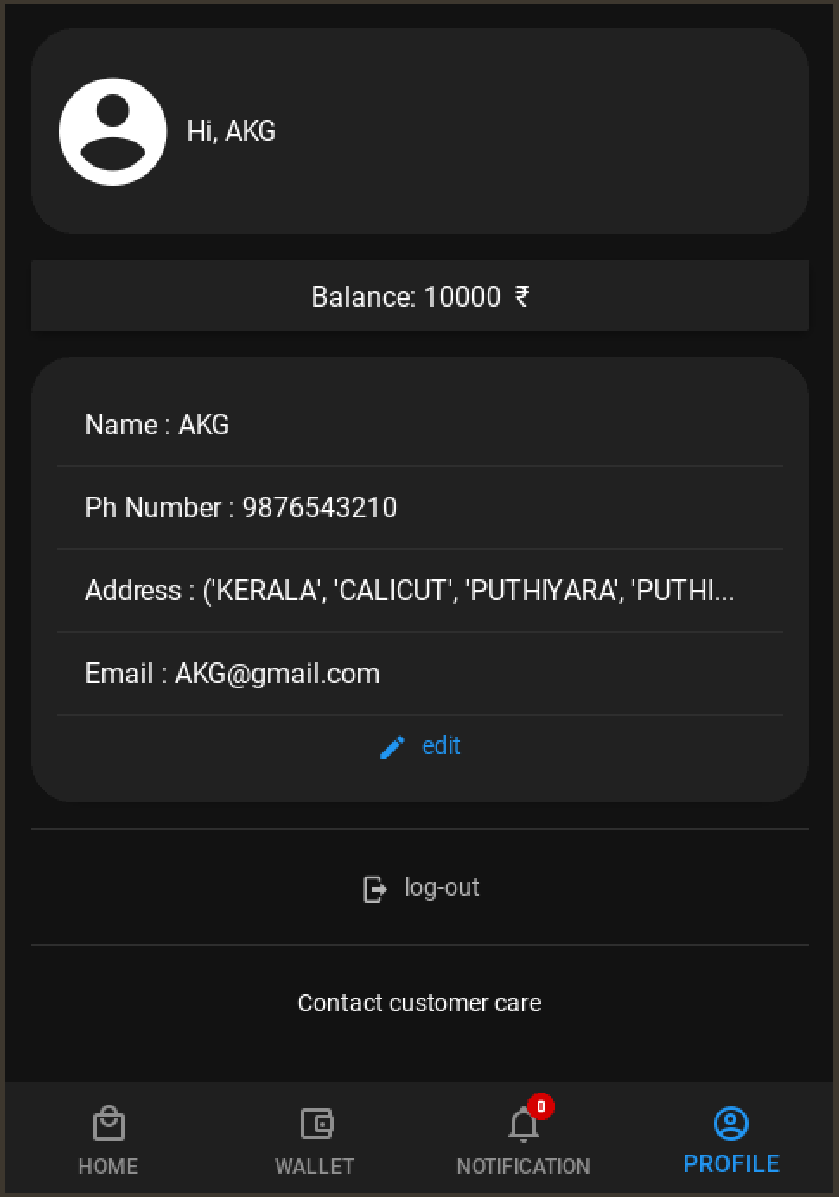
AbcStores : SPECIAL10 pending

HOME

WALLET

NOTIFICATION

PROFILE



Hi, AKG

Balance: 10000 ₹

Name : AKG

Ph Number : 9876543210

Address : ('KERALA', 'CALICUT', 'PUTHIYARA', 'PUTHI...

Email : AKG@gmail.com

edit

log-out

Contact customer care

HOME

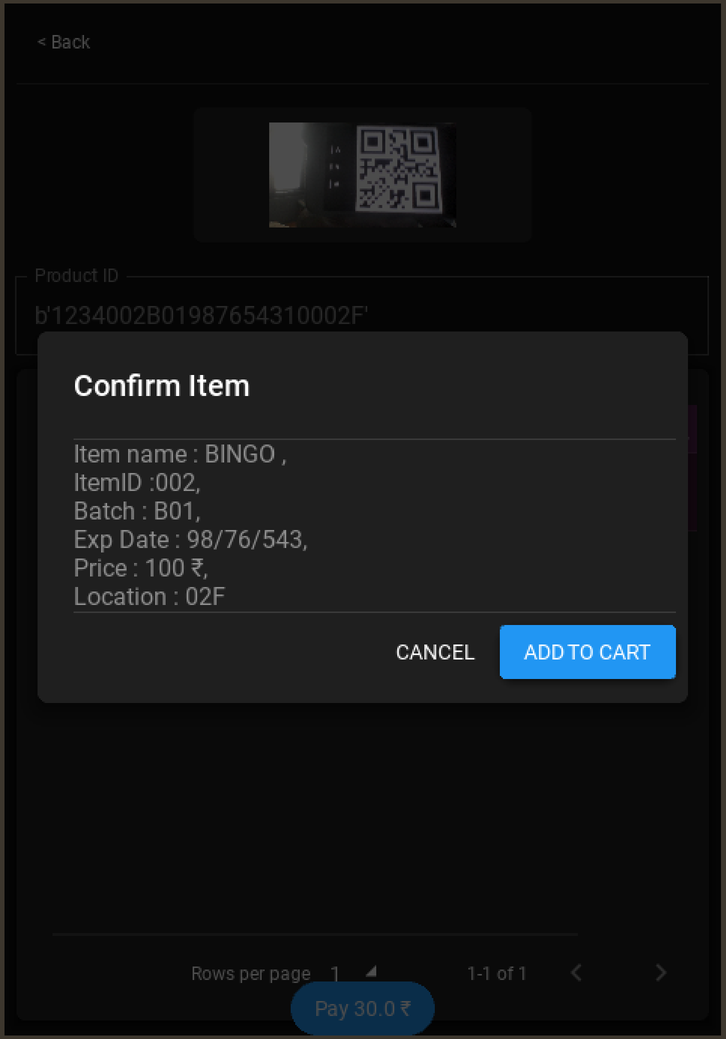
WALLET

NOTIFICATION

PROFILE

QR-CODE SCANNER

- The QR-code reader reads the data from the QR-code that sticks on the product.
- The data contain Product name, ID, batch number, expire date, price, location, store id, etc.
- Each QR-code is unique it self.



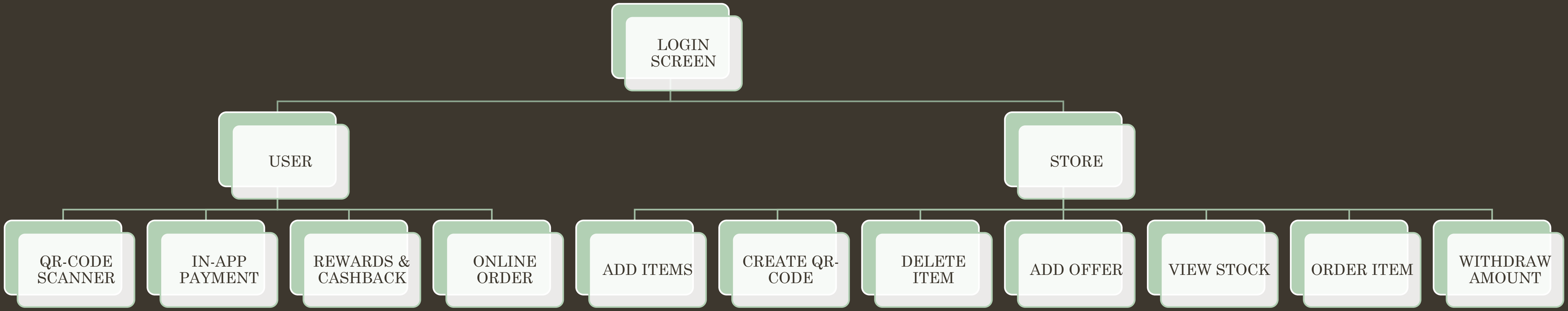
Product ID

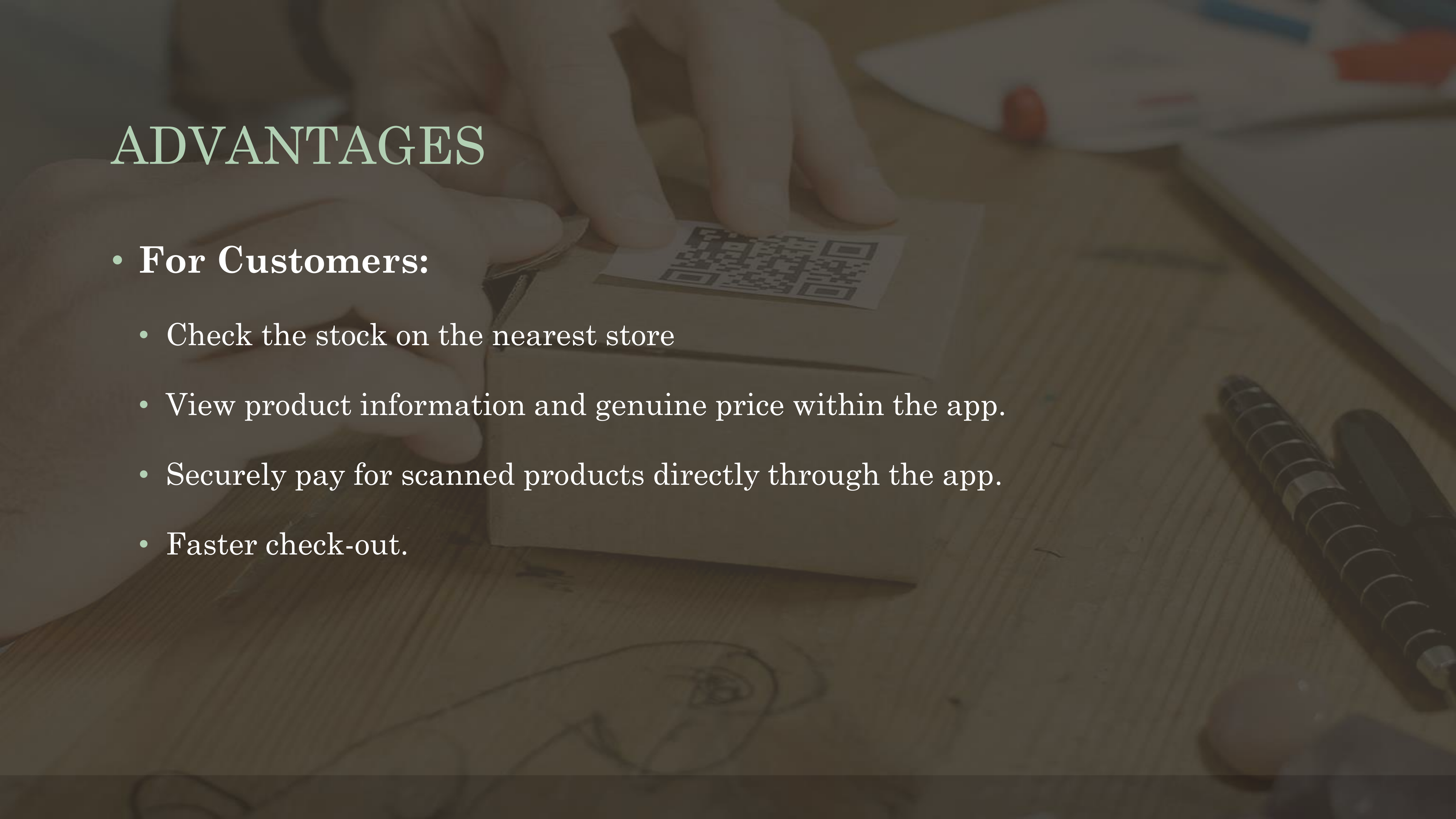
b'1234002B01987654310002F'

Confirm Item

Item name : BINGO ,
ItemID :002,
Batch : B01,
Exp Date : 98/76/543,
Price : 100 ₹,
Location : 02F

FLOW DIAGRAM





ADVANTAGES

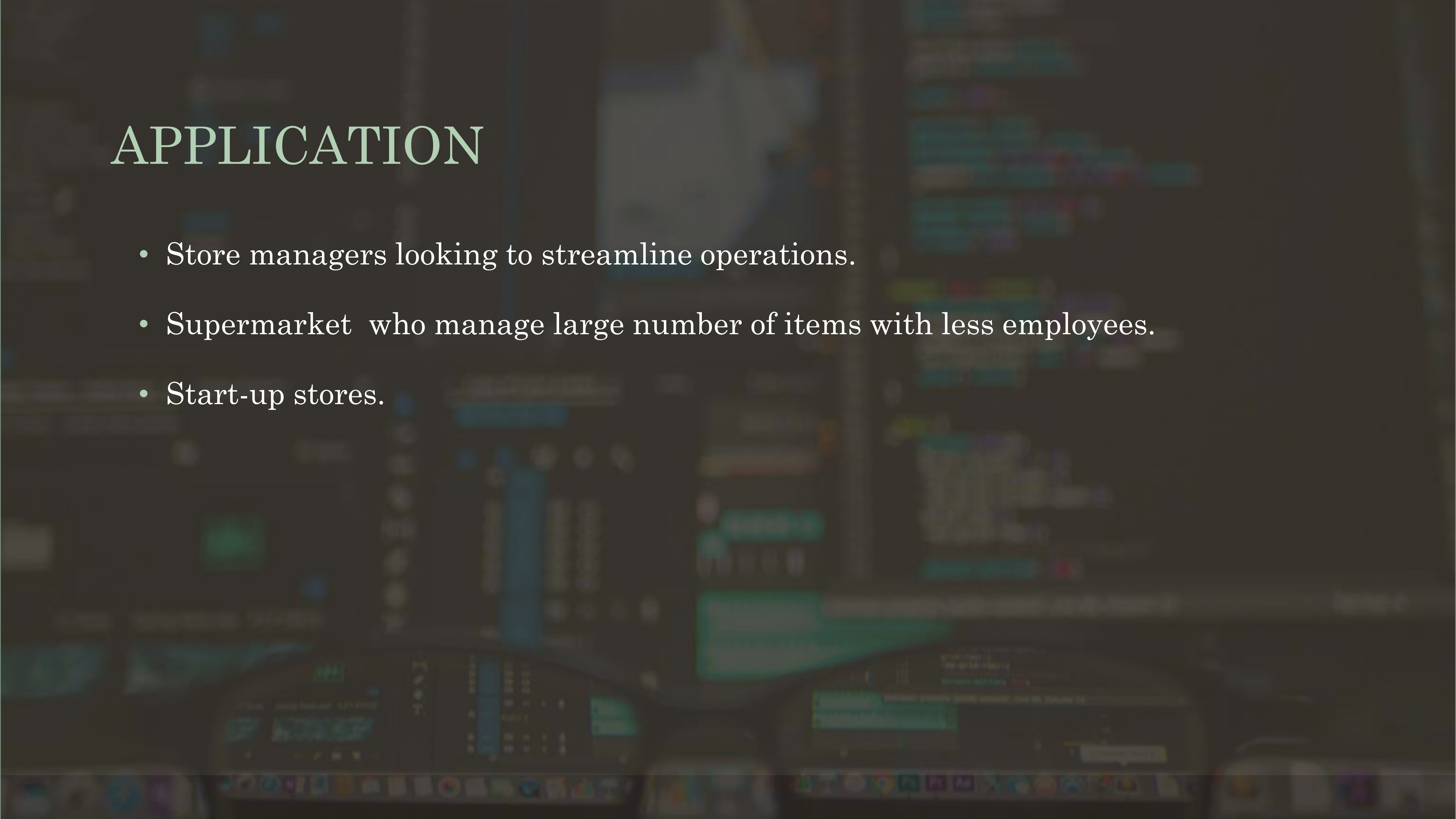
- **For Customers:**
  - Check the stock on the nearest store
  - View product information and genuine price within the app.
  - Securely pay for scanned products directly through the app.
  - Faster check-out.

ADVANTAGES

- **For Store Managers:**
  - Improve productivity and efficiency of store.
  - Easily add new products and set prices.
  - Generate unique QR codes for each product.
  - Track inventory levels in real-time for efficient stock management.
  - Manage existing products (edit details or delete).

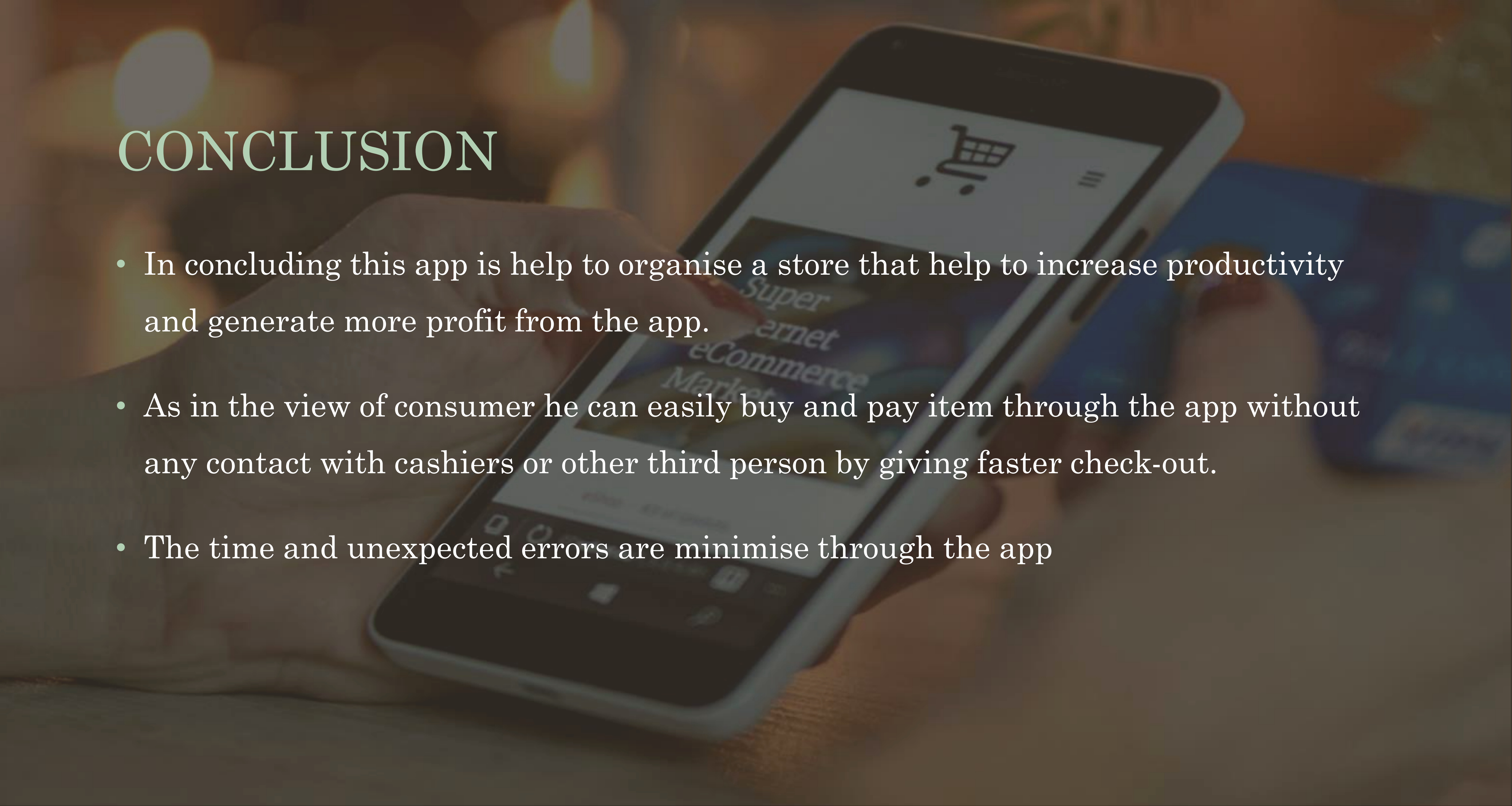
DIS-ADVANTAGES

- Limited availability
- Technical dependencies
- Security concern
- Privacy issue
- Learning Curve
- Stable internet connectivity
- Need to paste QR-Code for each product.



APPLICATION

- Store managers looking to streamline operations.
- Supermarket who manage large number of items with less employees.
- Start-up stores.



CONCLUSION

- In concluding this app is help to organise a store that help to increase productivity and generate more profit from the app.
- As in the view of consumer he can easily buy and pay item through the app without any contact with cashiers or other third person by giving faster check-out.
- The time and unexpected errors are minimise through the app

REFERENCE

- kivy.org
- Kivy-MD 1.9.1
- Kivy-MD-git
- python.org

- Ananthu S
- Aswanth S
- Anugrah A
- Rishan
- Adith
- Adithyan C P
- Nihal

THANK YOU