**ТАШКЕНТСКИЙ МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ ОБРАЗОВАНИЯ**



**КУРСОВАЯ РАБОТА**

**по дисциплине**

**«Разработка программного обеспечения»**

**на тему:**

**Интернет-магазин мебели**

*[Название темы]*

|  |  |
| --- | --- |
| ВЫПОЛНИЛ: |  |
| [Рахмонқулов Ахрор] |  |
| Студент 1 курса группы RI1-01  Направления информационные технологии |  |

Ташкент 2022 г.

**ОГЛАВЛЕНИЕ**

1. Постановка задач
2. Канбан таблица
3. Функциональная структура
4. UML диаграмма
5. Описание классов
6. Исходный код
7. Демонстрация кода QR code
8. QR code на гитхаб

ПОСТАНОВКА ЗАДАЧИ ПРИЛОЖЕНИЯ

1.Проблемы

Узбекский рынок мебелей относительно молод и имеет большой потенциал развития, так как растет число корпоративных потребителей, воспринимающих мебелей как естественную часть своей корзины. Ассортимент, пользующийся широким спросом в Узбекистане, составляет примерно 1 – 3 тысячи наименований. Западные компании в среднем имеют ассортимент в 5 – 7 тысяч позиций. По сравнению с западными странами спрос остается низким, но уровень корпоративной офисной культуры в Узбекистане растет и это стимулирует рост и развитие рынка.

2.Сценарий.

Последнее время я задаю один и тот же проверочный вопрос продавцам мебели: «Вы заметили, что покупатель сейчас к вам пришел более требовательный?»  
  
И все отвечают, что это так.  
«А знаете почему?», – продолжаю я.  
  
А вы знаете?  
Если пока нет, то читайте дальше - отвечу.  
  
Если сравнить те компании, которые за последние два года увеличили объем продаж, и те, которые показывают отрицательные результаты, можно увидеть одно, но очень яркое отличие.  
  
Первые – либо уже изменили, либо находятся в процессе подстройки бизнес-процессов под новую модель потребления покупателя мебели.  
Вторые - всё ещё ждут, когда рынок вернется на круги своя.  
  
Не вернется.  
  
И вот почему.  
Покупатель изменился.  
Он стал экономнее – причем во всем: и в том, что касается денег, и в том, что касается времени.  
  
Новая модель поведения покупателей мебели вынуждает мебельные компании менять свои бизнес-процессы.  
  
При этом приходится учитывать , по которым сегодня выбирают мебель в Эконом, Среднем и Премиум сегментах.

3 ТРЕБОВАНИЯ К ИНТЕРНЕТ-МАГАЗИНУ

3.1 Требования к программе

3.1.1 Требования к структуре и функциональным характеристикам Интернет-магазин должен представлять собой информационную структуру, доступную в сети Интернет. Интернет-магазин должен состоять из взаимосвязанных разделов с четко разделенными функциями. Интернет-магазин должен обеспечивать возможность выполнения перечисленных ниже функций: – просмотр информации о пользователях, товарах и заказах, находящихся в базе данных; – изменение и удаление информации о пользователях и товарах; – изменение информации о заказах.

3.1.2 Требования к пользователю Для эксплуатации Web-интерфейса системы управления сайтом от пользователя не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным Web-браузером.

3.1.3 Требования к разграничению доступа Информация, размещаемая в Интернет-магазине, является общедоступной. Пользователей сайта можно разделить на 2 части в соответствии с правами доступа: Пользователи и администратор. Пользователи имеют доступ только к общедоступной информации. Доступ к административной части имеют пользователи с правами администратора. Администратор может: добавлять или удалять пользователей. Доступ к административной части должен осуществляться с использованием уникального логина и пароля. Логин выдается администратором Web-приложения. Для обеспечения защиты от несанкционированного доступа к административной части при составлении паролей рекомендуется придерживаться следующих правил:

1. Длина пароля должна быть не менее 8 символов;

2. Пароль должен состоять из цифр и латинских букв в разных регистрах;

3.Пароль не должен являться словарным словом или набором символов, находящихся рядом на клавиатуре.

3.2 Требования к функциям, выполняемым Интернет-магазином

3.2.1 Обязательные функции Интернет-магазин должен обеспечивать выполнение следующих функций: – содержать макет сайта, с помощью которого пользователь сможет использовать ресурсы магазина; – содержать постраничную навигацию по сайту системы; – осуществлять поиск товаров, по введенному пользователем запросу; – содержать список пользователей, получивших свой заказ с указанием дат получения;

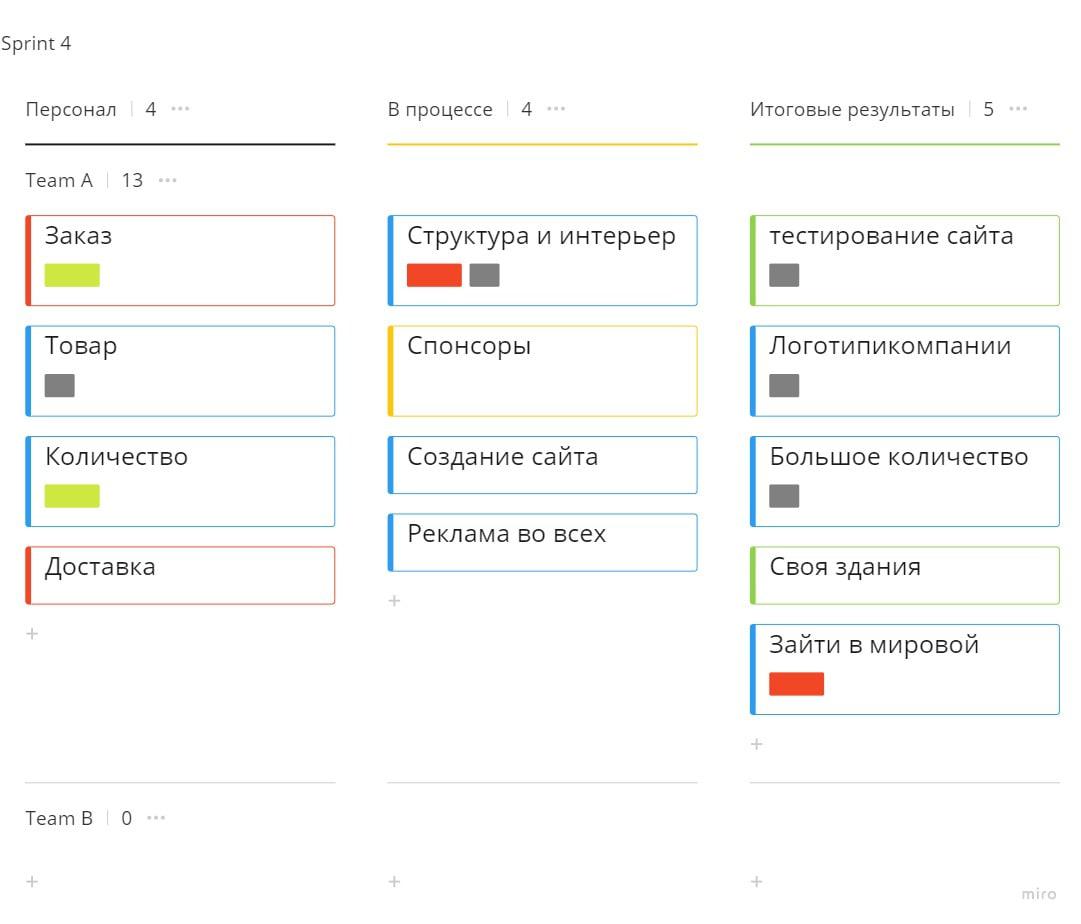
3.3 Требования к интерфейсу пользователя: Пользователь должен иметь возможность: – зайти на Web-сайт информационной системы; – чётко понимать, каким образом необходимо вести заполнение формы для поиска товара; – отправлять запросы для получения информации о товарах и оперативно получать результаты этих запросов.

4. Целевая среда Приложение должно быть продемонстрировано на Python.

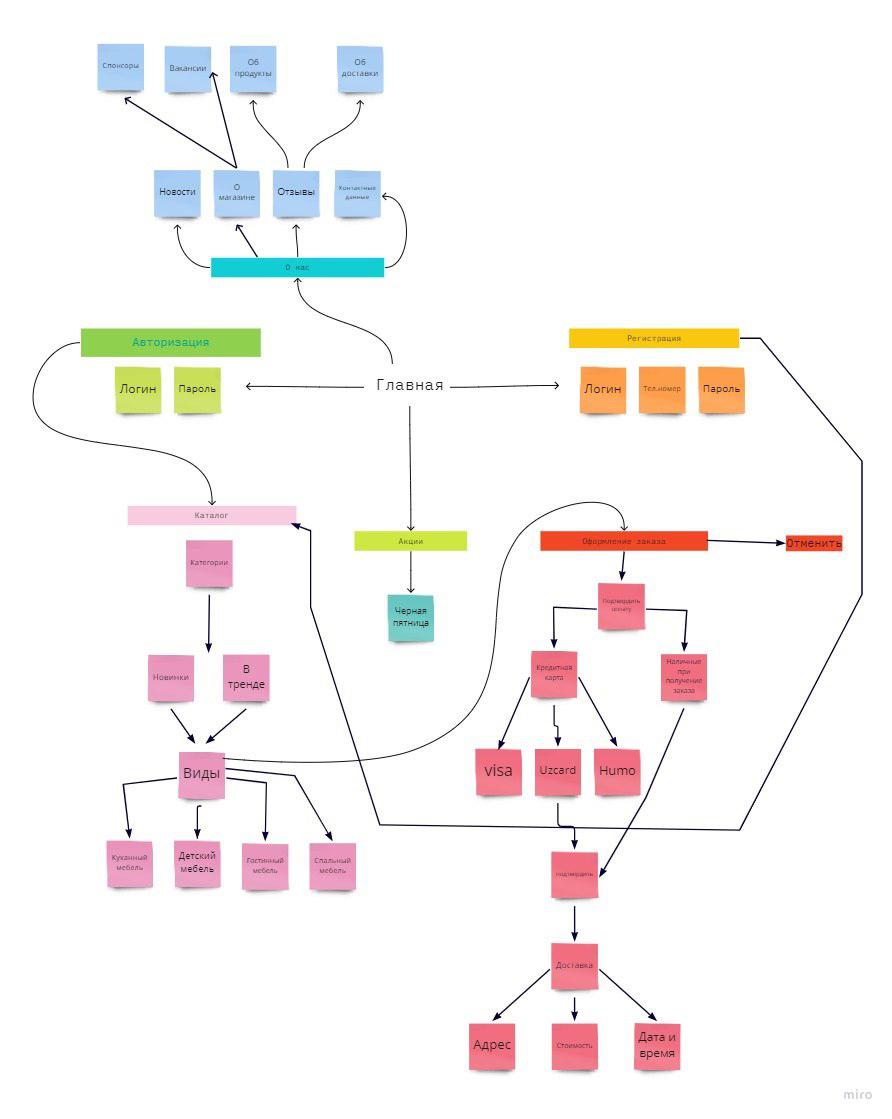
5. Документация • Документ с анализом требований (RAD - Requirements Analysis Document) • Проектный документ системы (SDD - System Design Document) • Исходный код под контролем версий, включая документацию по исходному коду.

**6.критерие, делающих интернет-магазин прибыльным**

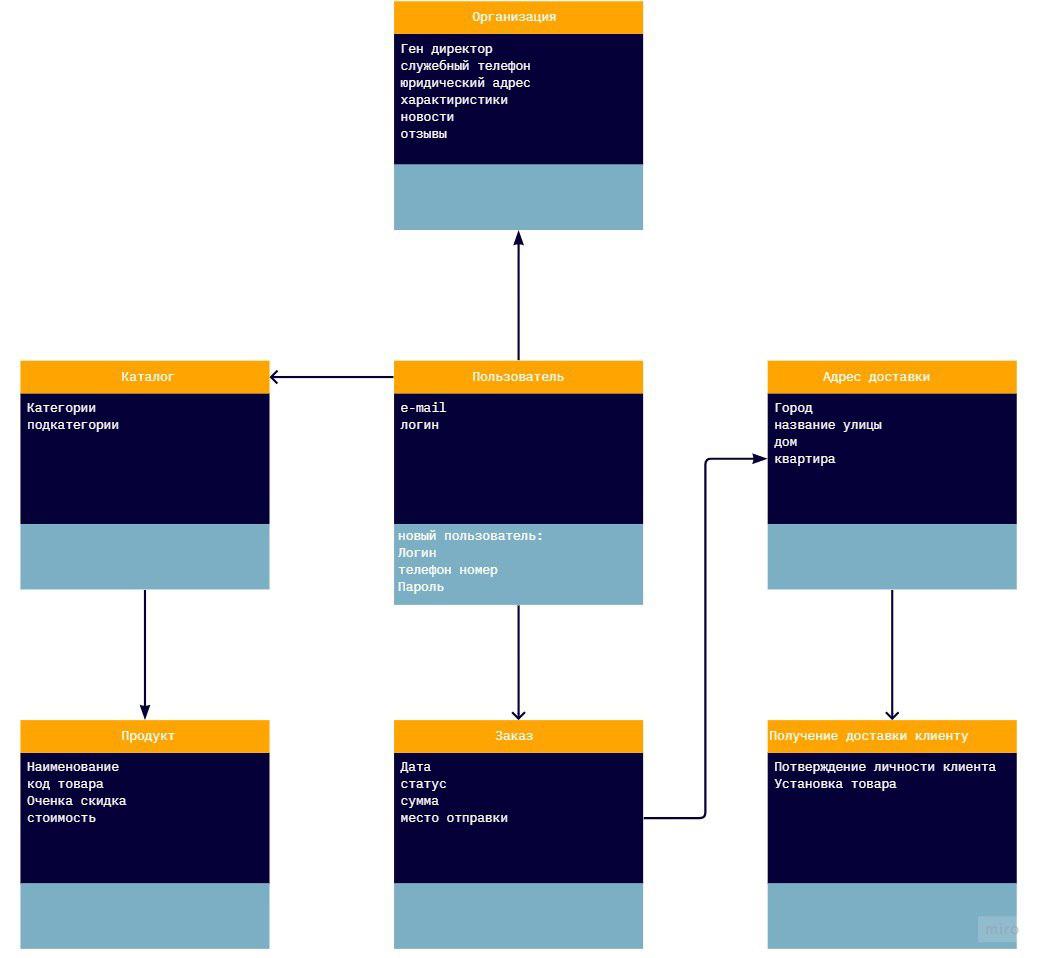
* Правильный дизайн ...
* Качественный поиск по сайту ...
* Покажите на главной странице свои ключевые товары ...
* Подробная информация о товаре ...
* Наличие отзывов, рекомендаций и обзоров товаров ...
* Призыв к действию ...
* Хорошая репутация компании ...
* Удобные способы оплаты заказа



Канбан таблица



Функциональная структура



UML диаграмма

Описание классов

MenuItem Класс Атрибута:

* Self.name
* Имя заказчика

-self.age

-Возраст клиента.например саша

-self.login

-логин заказчика

-self.password

-пароль заказчика

-self.location

-локация доставки

-self.time

-время доставки

-self.phone

-телефон номер заказчика

# Frame creation inmaster  
self.left = Frame(master, width=800, height=720, bg='orange')  
self.left.pack(side=LEFT)  
self.right = Frame(master, width=800, height=720, bg='steelblue')  
self.right.pack(side=RIGHT)  
  
self.heading = Label(self.left, text="Закозать товар", font=('arial 37 bold'), fg='green',  
 bg='orange')  
self.heading.place(x=0, y=0)  
  
##custemers name  
self.name = Label(self.left, text="Имя клиента", font=('arial 18 bold'), fg='green', bg='orange')  
self.name.place(x=50, y=200)  
# age  
self.age = Label(self.left, text="возраст", font=('arial 18 bold'), fg='green', bg='orange')  
self.age.place(x=50, y=250)  
# login  
self.age = Label(self.left, text="логин", font=('arial 18 bold'), fg='green', bg='orange')  
self.age.place(x=50, y=250)  
# Password  
self.gender = Label(self.left, text="пароль", font=('arial 18 bold'), fg='green', bg='orange')  
self.gender.place(x=50, y=300)  
# address  
self.location = Label(self.left, text="локация", font=('arial 18 bold'), fg='green', bg='orange')  
self.location.place(x=50, y=350)  
# appointment  
self.time = Label(self.left, text="время", font=('arial 18 bold'), fg='green', bg='orange')  
self.time.place(x=50, y=400)  
# phone  
self.phone = Label(self.left, text="телефон", font=('arial 18 bold'), fg='green', bg='orange')  
self.phone.place(x=50, y=450)

Исходный код

from tkinter import \*

import tkinter.messagebox as tkm

# from tkinter import Frame

# from tkinter import Label

import sqlite3

# databse connection

conn = sqlite3.connect('database.db')

print("Connected")

c = conn.cursor()

# the GUI window using tkinter

class Mainfile:

def init(self, master):

self.master = master

# Frame creation inmaster

self.left = Frame(master, width=800, height=720, bg='orange')

self.left.pack(side=LEFT)

self.right = Frame(master, width=800, height=720, bg='steelblue')

self.right.pack(side=RIGHT)

self.heading = Label(self.left, text="Закозать товар", font=('arial 37 bold'), fg='green',

bg='orange')

self.heading.place(x=0, y=0)

##custemers name

self.name = Label(self.left, text="Имя клиента", font=('arial 18 bold'), fg='green', bg='orange')

self.name.place(x=50, y=200)

# age

self.age = Label(self.left, text="возраст", font=('arial 18 bold'), fg='green', bg='orange')

self.age.place(x=50, y=250)

# login

self.age = Label(self.left, text="логин", font=('arial 18 bold'), fg='green', bg='orange')

self.age.place(x=50, y=250)

# Password

self.gender = Label(self.left, text="пароль", font=('arial 18 bold'), fg='green', bg='orange')

self.gender.place(x=50, y=300)

# address

self.location = Label(self.left, text="локация", font=('arial 18 bold'), fg='green', bg='orange')

self.location.place(x=50, y=350)

# appointment

self.time = Label(self.left, text="время", font=('arial 18 bold'), fg='green', bg='orange')

self.time.place(x=50, y=400)

# phone

self.phone = Label(self.left, text="телефон", font=('arial 18 bold'), fg='green', bg='orange')

self.phone.place(x=50, y=450)

# Since we have created the labels for all the info

# Now we need to create lables to accept entries

self.nameEntry = Entry(self.left, width=50)

self.nameEntry.place(x=250, y=210)

self.ageEntry = Entry(self.left, width=10)

self.ageEntry.place(x=250, y=260)

self.genderEntry = Entry(self.left, width=20)

self.genderEntry.place(x=250, y=310)

self.locEntry = Entry(self.left, width=50)

self.locEntry.place(x=250, y=360)

self.timeEntry = Entry(self.left, width=20)

self.timeEntry.place(x=250, y=410)

self.phoneEntry = Entry(self.left, width=20)

self.phoneEntry.place(x=250, y=460)

# creating a button

self.done = Button(self.left, text='заказать', width=20, height=3, bg='white',

command=self.addAppointment)

self.done.place(x=310, y=560)

def addAppointment(self):

# print("AllSet this was for checking")

# get() for user inputs

# somewhat similar to C++/object oriented programming

self.x1 = self.nameEntry.get()

self.x2 = self.ageEntry.get()

self.x3 = self.genderEntry.get()

self.x4 = self.locEntry.get()

self.x5 = self.timeEntry.get()

self.x6 = self.phoneEntry.get()

# To check if the above thoings are working or not....

# Dummy Print

# print(self.x1)

# print(self.x2)

# print(self.x3)

# print(self.x4)

# print(self.x5)

# print(self.x6)

if self.x1 == '' or self.x2 == '' or self.x3 == '' or self.x4 == '' or self.x5 == '' or self.x6 == '':

# print("All boxes are cumpolsury")

tkm.showinfo("ERROR!!!", "All boxes are cumpulsory")

else:

print("ALL INPUTS Done")

# database transfer of details

sql = "insert into 'appointments' (name,age,login,password,location,scheduled\_time,phone) values(?,?,?,?,?,?)"

c.execute(sql, (self.x1, self.x2, self.x3, self.x4, self.x5, self.x6))

conn.commit()

print("database retrieval successful!!!")

# Objects

root = Tk()

b = Mainfile(root)

# Dimensions of the window are being set

# root.geometry('%dx%d+%d+%d' % (width, height, x, y))----got to know from StackOverflow

root.geometry("1200x720+0+0")

root.resizable(False, False)

root.mainloop()

Демонстрация кода

<http://qrcoder.ru/code/?https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DOlarPuXlWqo&4&0>



QR на гитхаб

<http://qrcoder.ru/code/?https%3A%2F%2Fgithub.com%2Fahrorrahmankulov%2Ftiue_SD&4&0>

