



Wedding Dress System-Group Report

B9CY100
Advanced
Programming
Techniques

Lecturer:

Paul Laird

By:

Aravinth Joshua santhi

Paramjeet Singh

Victor Obi Okeke

Contents

Introduction:	3
Project objectives:.....	3
Use case Functions:.....	3
Implementation of the project:	6
Technology used:	6
Python GUI with Tkinter.....	6
Database:	6
Application structure:	7
Contributions:	7
Git Repository:	8
Testing the application and output:.....	8
References:	12

Introduction:

There are many vendors those provide Wedding dress on rent. Offline booking system is having many overheads like timely stock not updated, customers have to come many times to the shop etc. To overcome these problems online system needs to be developed.

Project objectives:

This project aim is to develop a web based online system for dress booking.

Use case Functions:

Wedding dress Rental System is a Web application that would allow its customer to buy wedding dress. While Admin can manage entire System.

Admin Follow these main functions:

- Login
- add_dress
- remove_dress
- cacle_booking
- fetch_dress
- fetch_booking
- fetch_customer
- Logout

Customer Follow these main functions:

- Sign_Up
- Login
- Choose_Dress
- Make_Booking
- Show_Booking

ER-modeling:

It is a data modeling technique used in software engineering to produce a conceptual data of an information system.

ERD Details

Entity:

- Customer
- Store
- Dress
- Dress_Category
- Stock
- Booking

In this ERD entity **Customer** contain all details about Customer, with help of this system customer can book any dress. All customer have a unique identity number, which is Customer_id in the table. Customer table have basic Attributes Such as Name, Gender, Age, Phone_No, Email, Address, Postal_Code, User_ID, Password. It has connectivity **1:M** from Booking.

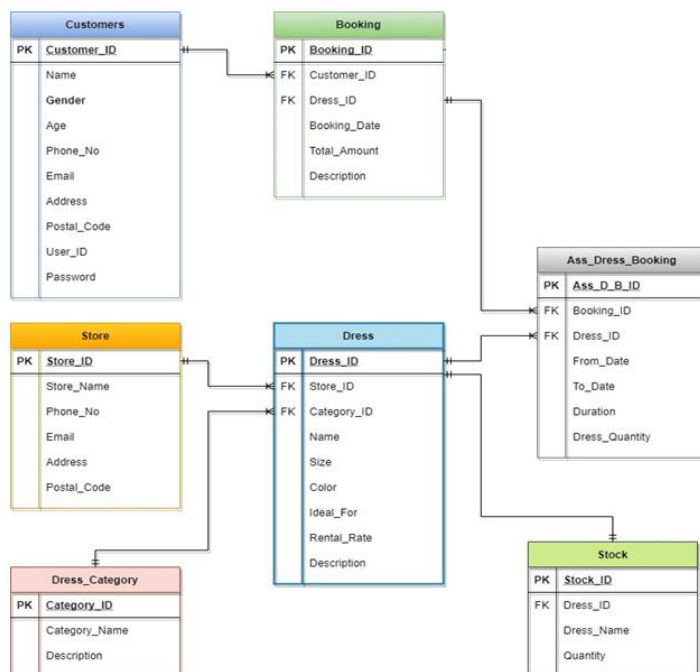
In this ERD second Entity is **Store**. All the details of Store are available in this table. All Stores have a unique identity number, which is Store_ID in the table. Store table have basic Attributes Such as Store_Name, Phone_No, Email, Address, Postal_code. It has connectivity **1:M** from Dress.

Here is another entity is **Dress _Category**. which stores information about Dress Category. Dress Category table has a unique number which is Category_ID. Dress_Category table have some attribute such has Category_Name, Description. It has connectivity **1:M** from Booking.

Another Entity is **Dress** which stores information about Dress. Dress has a unique number which is Dress_ID. Dress have many attribute such as Store_ID, Category_ID, Name, Size, Color, Ideal_for, Rental_Rate, Description. It has connectivity **1:M** from store and Dress_Category, **1:1** from stock and **M:M** from Booking.

Here is another entity **Stock**, It manage all information about Stock. Stock table has a unique number which is Stock_id with help of this unique key we can easily find any Stock details. Stock have many attribute such as Dress_ID, Dress_Name, Quantity. It has connectivity **1: 1** from Dress.

Another Entity is **booking**, it manage all information about Booking. Booking have a unique number which is Booking_ID with help of this unique key we can easily find any Booking details. Booking have many attribute such as Booking_ID, Customer_ID, Dress_ID, Booking_Date, Total_Amount, and Description. Booking has connectivity **1:M** Customer, **1:1** from Payments and **M:M** from Dress.



Relation Model

- **Customer**(Customer_ID, Name, Gender, Age, Phone_No, Email, Address, Postal_Code, User_ID, Password)
- **Store**(Store_ID, Store_Name, Phone_No, Email, Address, Postal_Code)
- **Dress_Category**(Category_ID, Category_Name, Description)
- **Dress**(Dress_ID, Store_ID*, Category_ID*, Name, Size, Color, Ideal_For, Rental_Rate, Description)
- **Stock**(Stock_Id, Dress_ID* Dress_Name, Quantity)
- **Booking**(Booking_ID, Customer_ID*, Dress_ID*, Booking_Date, Total_Amount, Description)

Implementation of the project:

Technology used:

Python GUI with Tkinter

In this assignment we have built a e-commerce application of online dress booking in python language where GUI was built in Tkinter and Sqlite3 is used for database purpose because it easy to use but can work for small work only and since our application won't be entertaining big request sqlite3 will be the great choice.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

To make GUI apps we need to perform below tasks:

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to act against each event triggered by the user.

Database:

To interact with data Sqlite3 is used as database. It is open source database and python can interact easily.

Application structure:

There are two types of users who will interact with our application and they are Admin, Customer. Admin has all high-level privilege where he can interact with the application can add or delete any dress also he has permission to delete any customer or cancel any running booking.

Customers will have common permission like he/she can book a dress for themselves and also can see the available dress in the store.

After he/she books a dress, they will be given a booking id which he can use to check the status of his booking.

Also, each customer will be given a unique customer id which acts as a primary key but since it is a long string you can use your Username to login.

Contributions:

It was a team effort which helped us to complete the project on time. With people managing their part-time jobs and still making time for the meeting everyday to work on the project was commendable. We all did brainstorm together to come up with this project and dedicated ourselves to get the project completed. It was a collaborative effort though the highlighted efforts of the team are mentioned below:

Aravinth Joshua Santhi: As the only member from programming background, he took initiative to coordinate with the team and initialized the first meeting. Henceforth he contributed in creating python functions for Main.Py including tkinter codes for the front end application. He assisted in creating functions for admin.py for the admin page and was involved in research and troubleshooting for the entirety of the project.

Paramjeet Singh: As an experienced professional, he was assigned responsibility to co-ordinate with the group, and he worked on the tables and database for the web application. He was heavily involved in the research and troubleshooting of the code, and he assisted in creating functions for the admin.py for the admin page. He created the framework for the project and was involved in the documentation of the assignments.

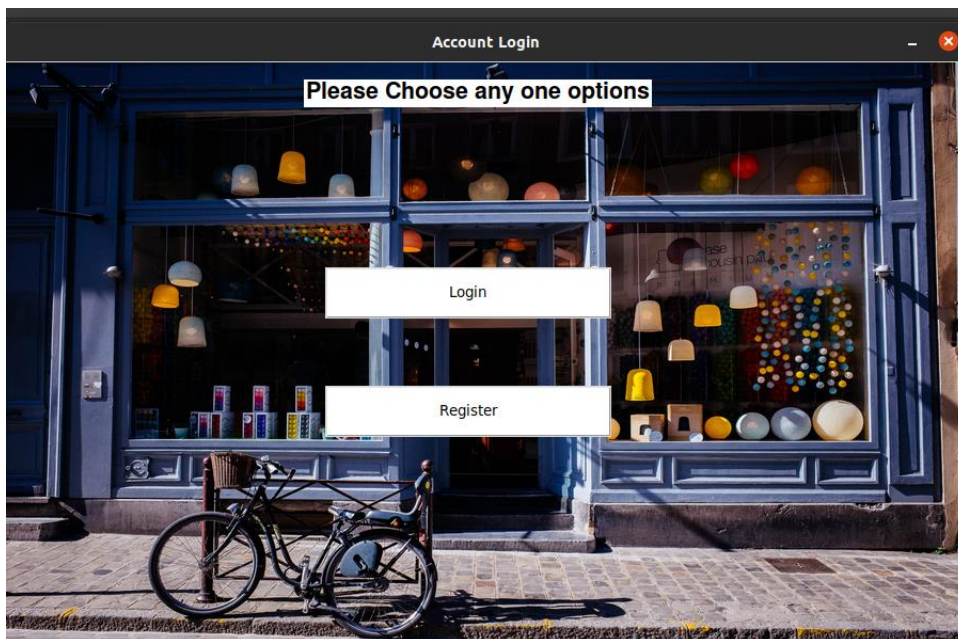
Victor okeke: He was a crucial member of our team and he assisted throughout in his capacity in creating the project. He was always available for the meetings and contributed heavily on the documentation and creating functions for the user.py for the customer web page. He also contributed on creating the framework for the database and was involved in research and troubleshooting of the code.

Git Repository:

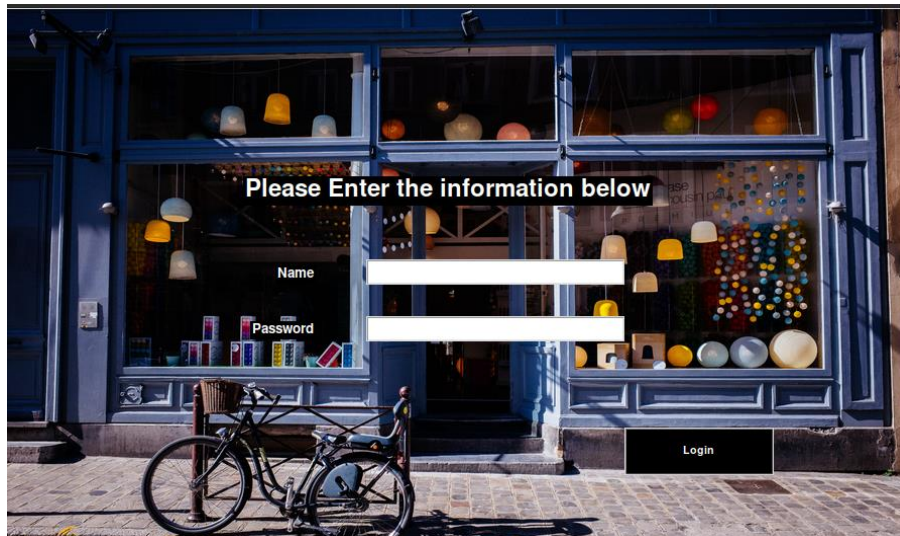
[singh1jeet/Projects: Assignment-Python CA02 \(github.com\)](https://github.com/singh1jeet/Projects: Assignment-Python_CA02)

Testing the application and output:

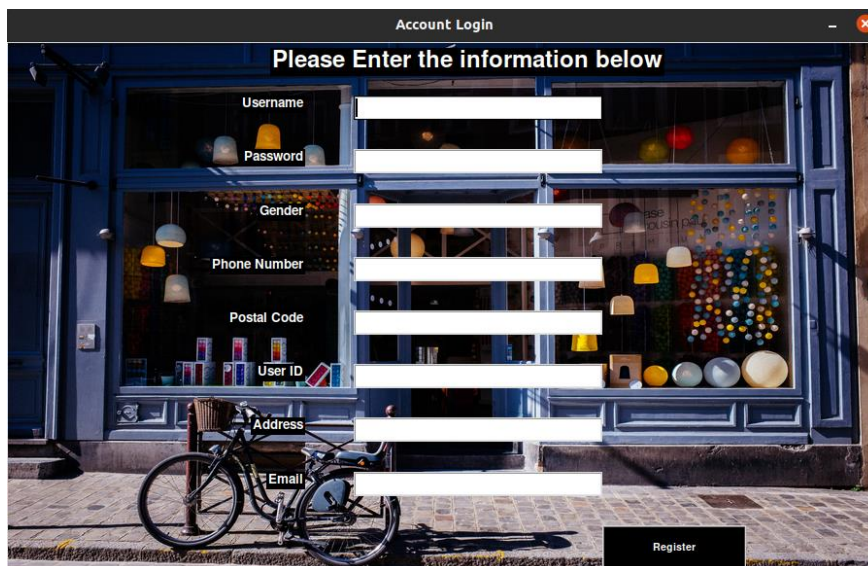
Main Screen of the application will look like below :



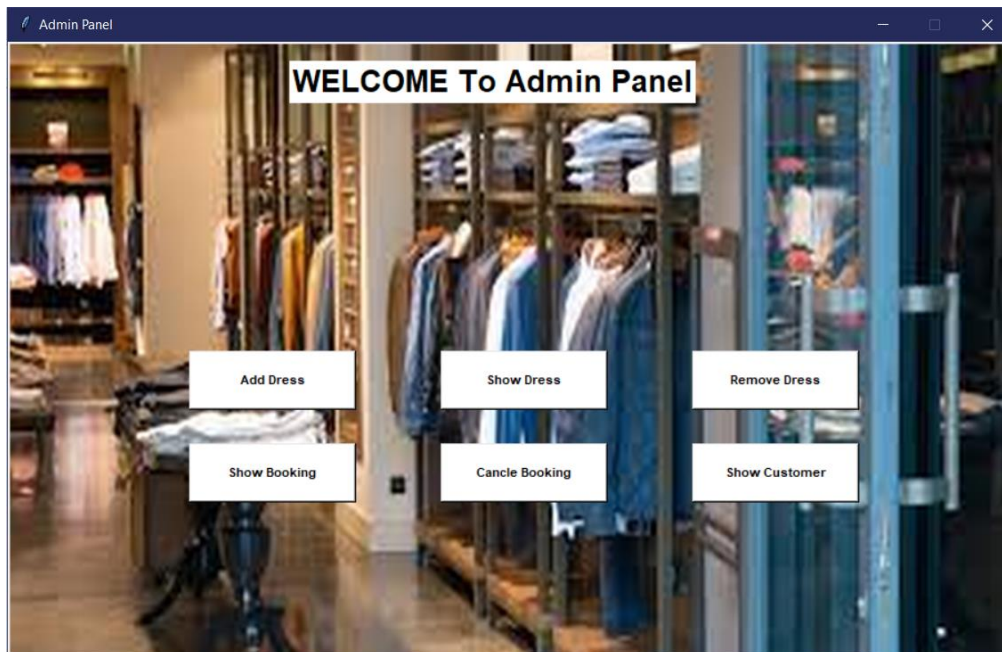
Login Screen: Using login page customer or admin can do login.



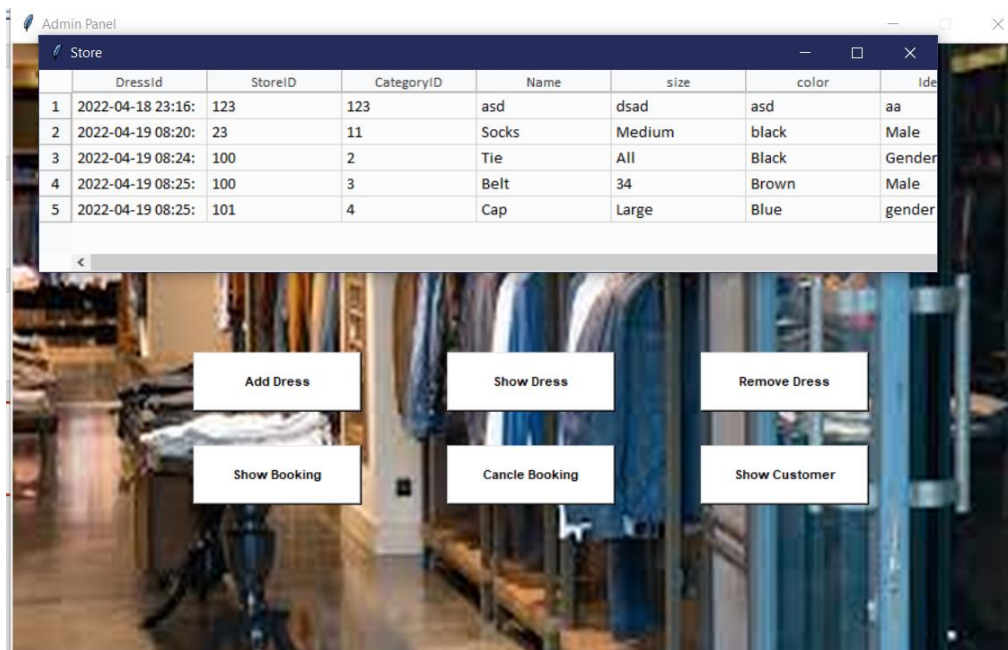
Register Screen: Using register page customer or admin can do register.



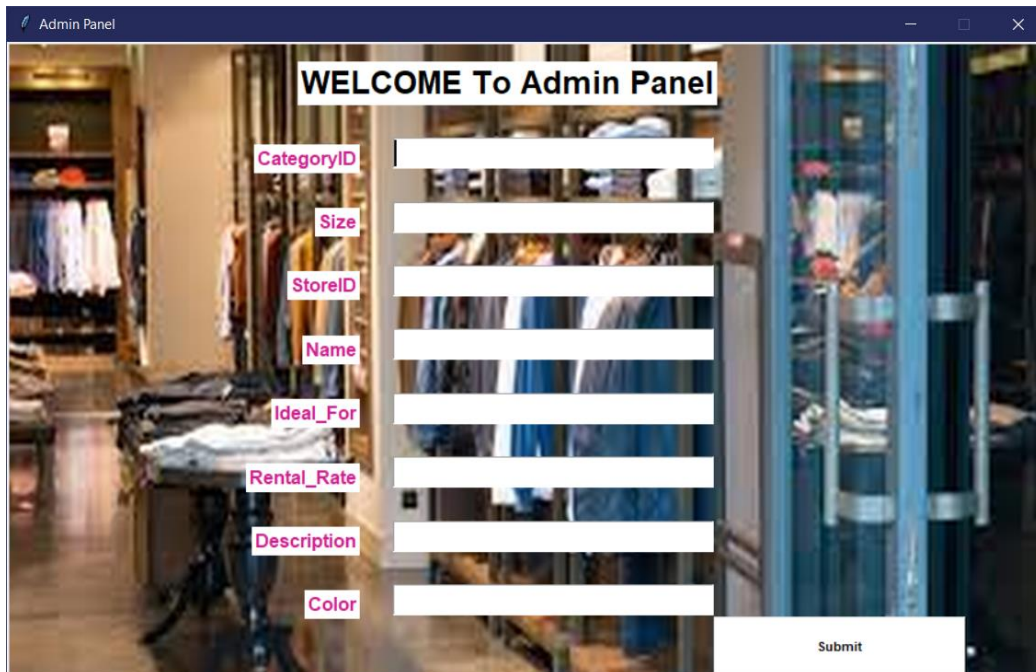
Admin Panel: admin panel is like extra privileges given to user and he can upload dress details etc.



Showing all available dress to admin

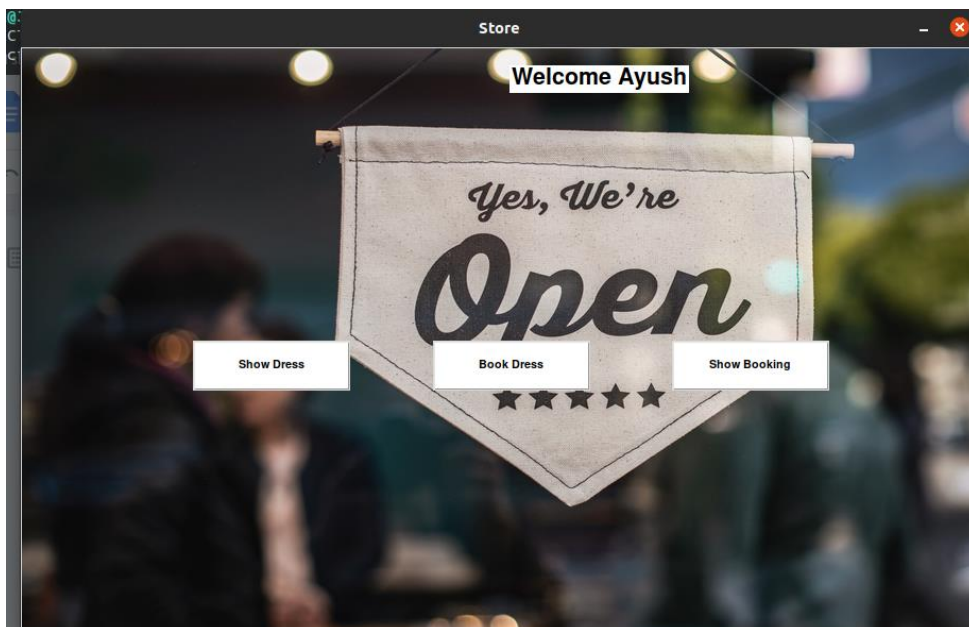


To add a dress using admin panel :



A screenshot of a web application's admin panel. The window title is "Admin Panel". The background is a blurred image of a clothing store interior. At the top, a black banner with white text reads "WELCOME To Admin Panel". Below this, there is a form with several input fields, each with a pink label to its left: "CategoryID", "Size", "StoreID", "Name", "Ideal_For", "Rental_Rate", "Description", and "Color". Each field is a white rectangular box. At the bottom right of the form area is a white button with the text "Submit".

Customer Screen



To login as admin type Username: - admin, and Password: - admin

References:

https://www.tutorialspoint.com/python/python_gui_programming.htm

<https://www.geeksforgeeks.org/python-gui-tkinter/>

<https://www.tutorialsteacher.com/python/create-gui-using-tkinter-python>