**Final Project Document**

**Brief Introduction about the website:**

This is a laundry management website called “Arial Cleaners” which offers number of services (Wash/dry-cleaning etc.) to its customers.

**Users and privileges:** There will be only one Log in page for the customers and the Admin. However, both admin and customer will have different views.

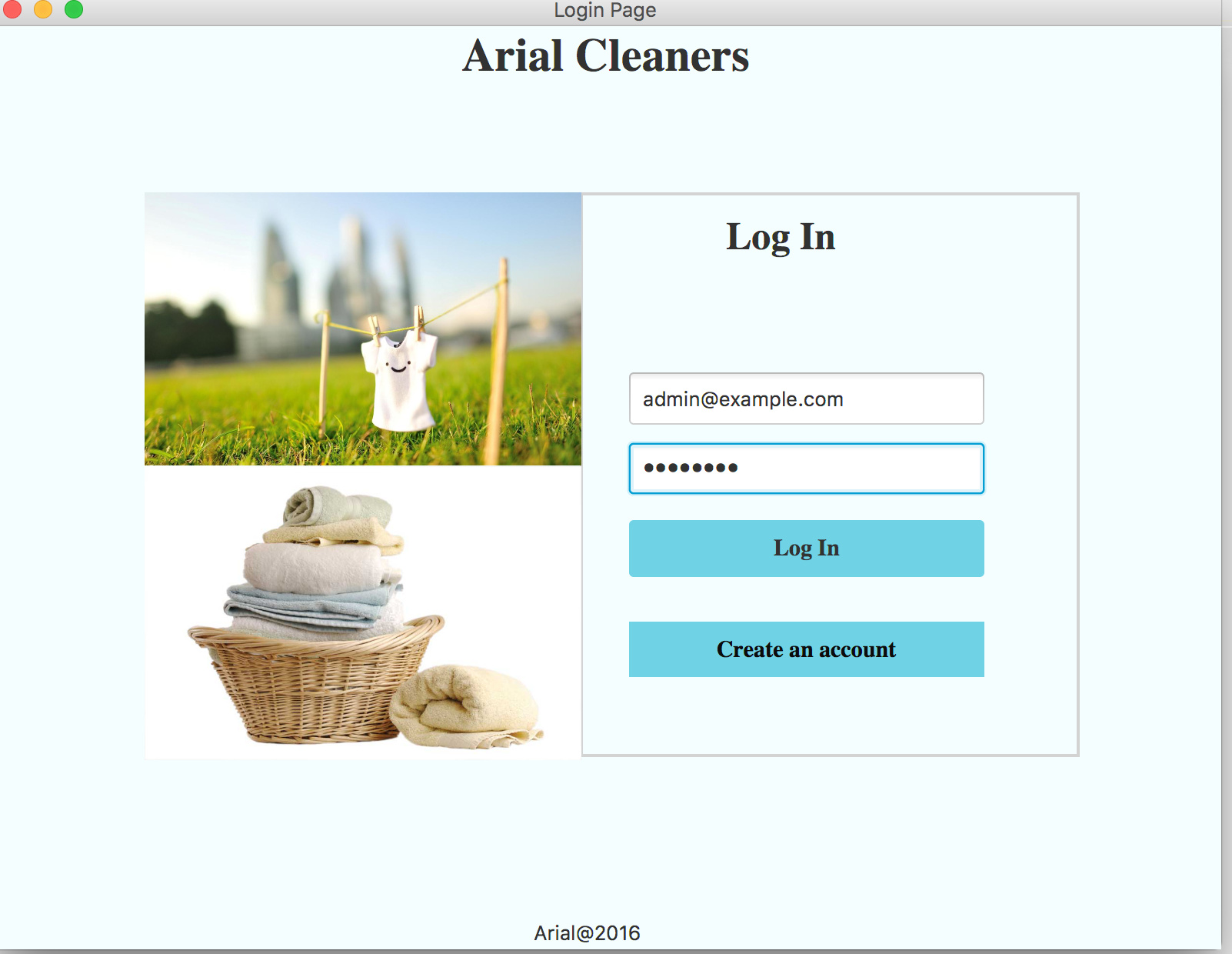
Admin can be created backend by SQL queries as a AdminUser role.

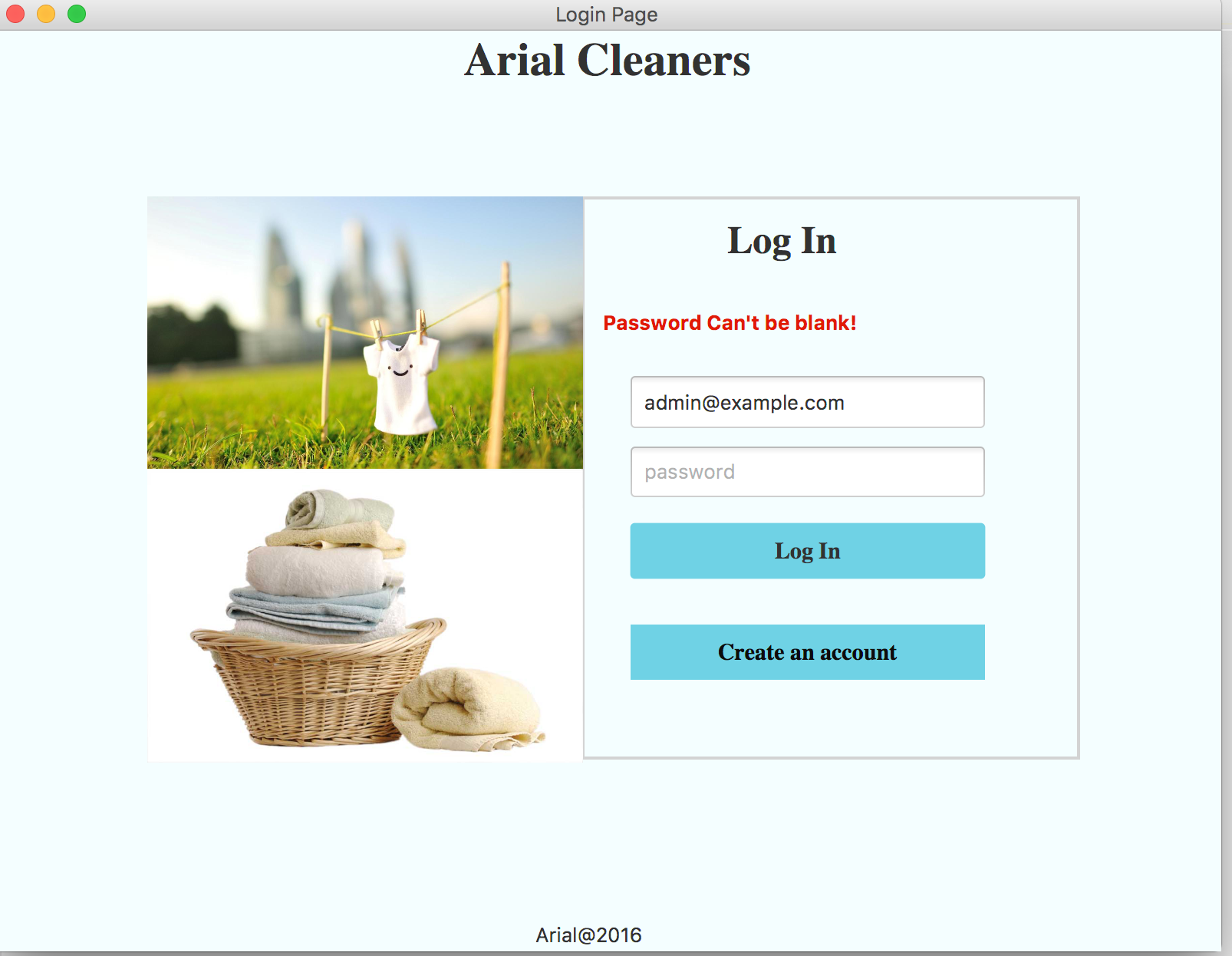
**Admin login credentials: Customer login credentials**

|  |
| --- |
| **Username:** [**admin@example.com**](mailto:admin@example.com) **tom.harley@gmail.com** |
| **Password: password. tomharley** |

**Admin Roles:**

1. Admin will log in with his username and password. Username and password validation is also done in the login page by displaying the error messages.



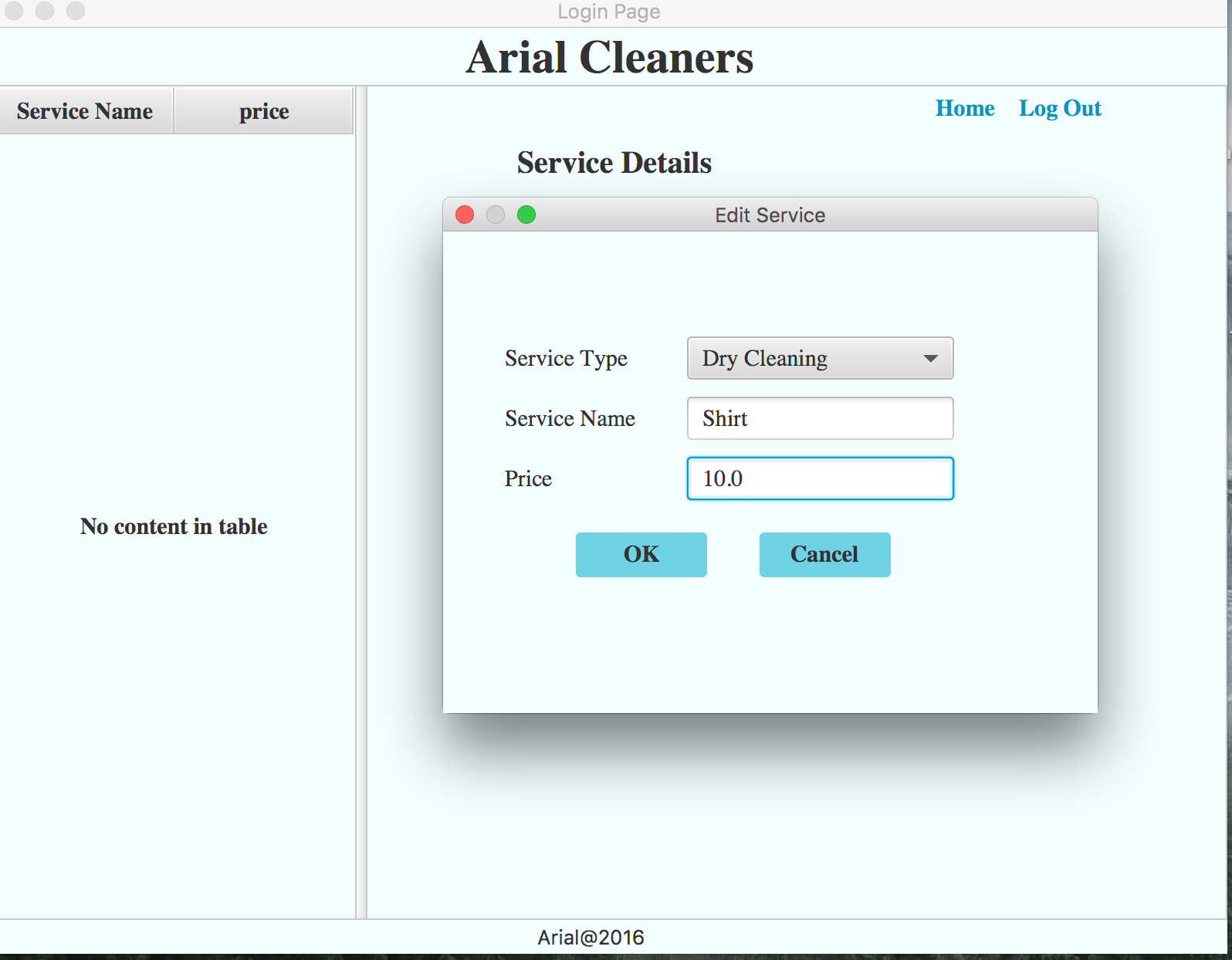


1. Once the admin logs in he will be directed to the Admin Dashboard where he can see three options (Shown in below screen shots):

* **Service Configuration:** Once he clicks on the Service Configuration tab, he will be able to ADD, EDIT and DELETE services.
* **Customers**: This tab will show all the customers of this website.
* **Orders**: This tab will give the admin a view of all the orders placed by the customers.



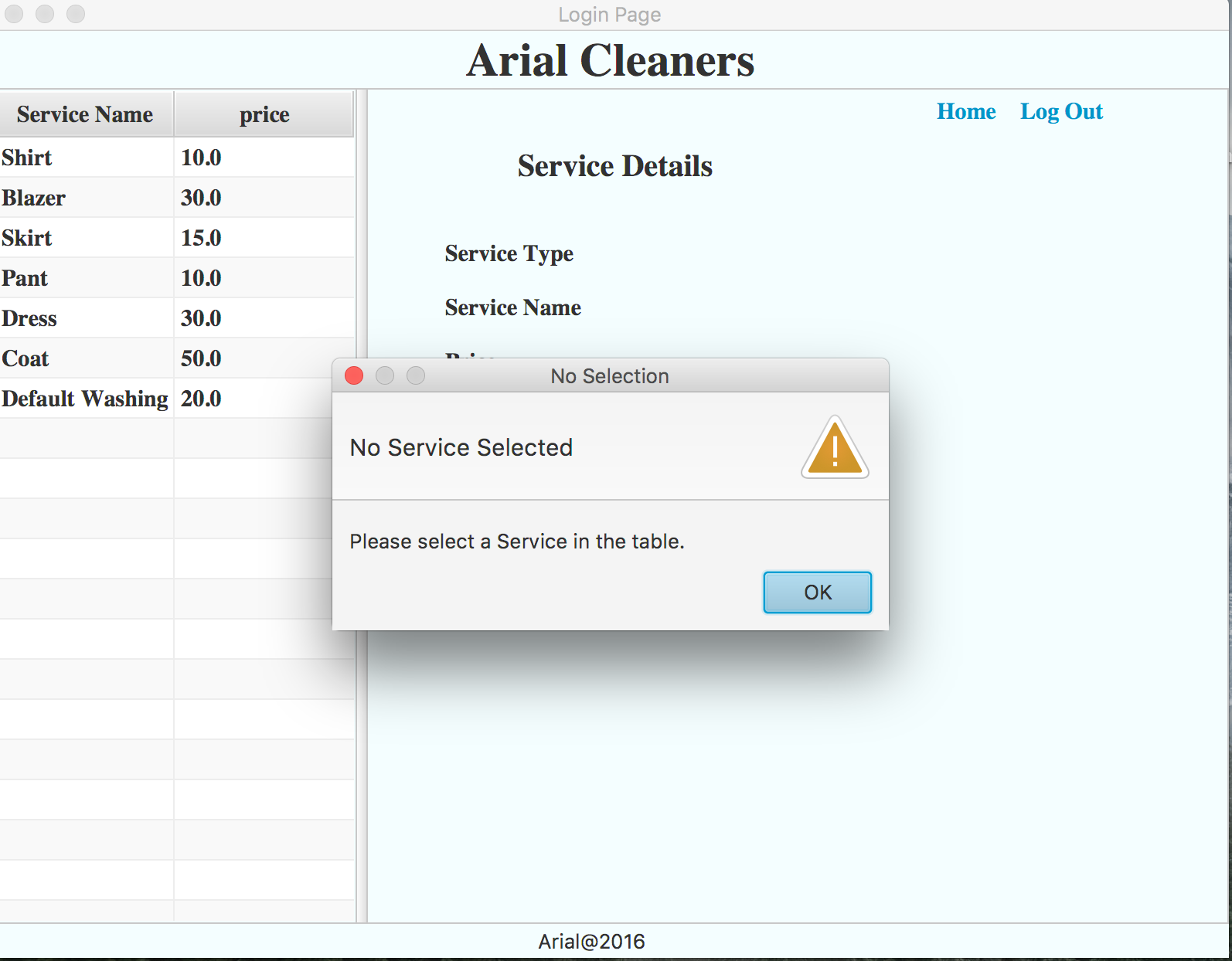
**Service Configuration Page:**



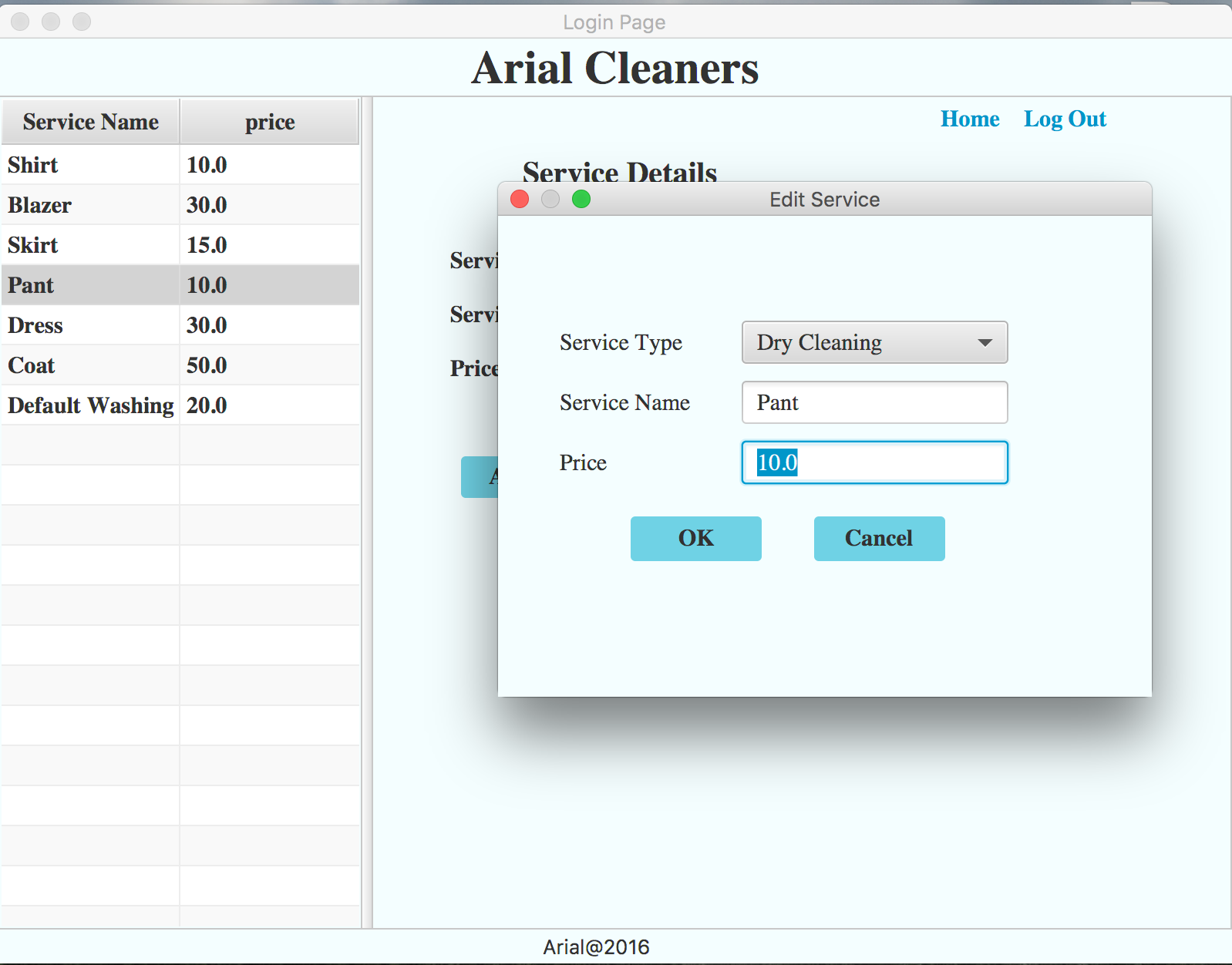
**Displays all the services which are added by the Admin**:



**Error message shown if the service is not selected:**



**Editing a service Item:**



**Customer tab: shows all the customers**

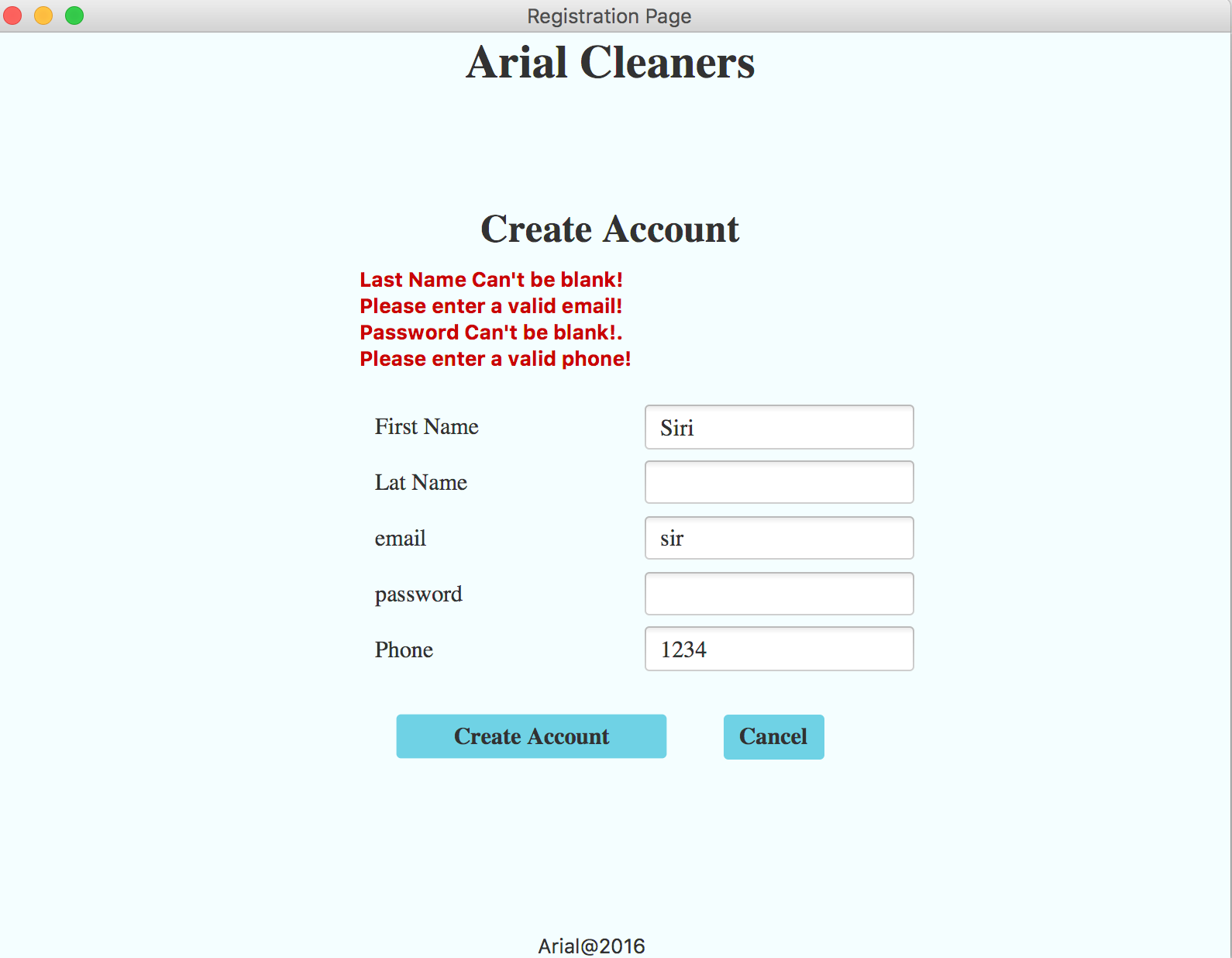


**Orders Tab: shows all the orders:**



**Customer Roles:**

1. An existing customer can log in by user his/her username or password, if the customer is new, he can click on the Register tab and fill out the required fields in the registration page, f the required fields are not initiated, then the system will prompt the error message to fill out the fields. Length validation for password is also shown below screen shots. Once he has successfully registered himself, he will again be taken back to the login page to log with his new username and password.

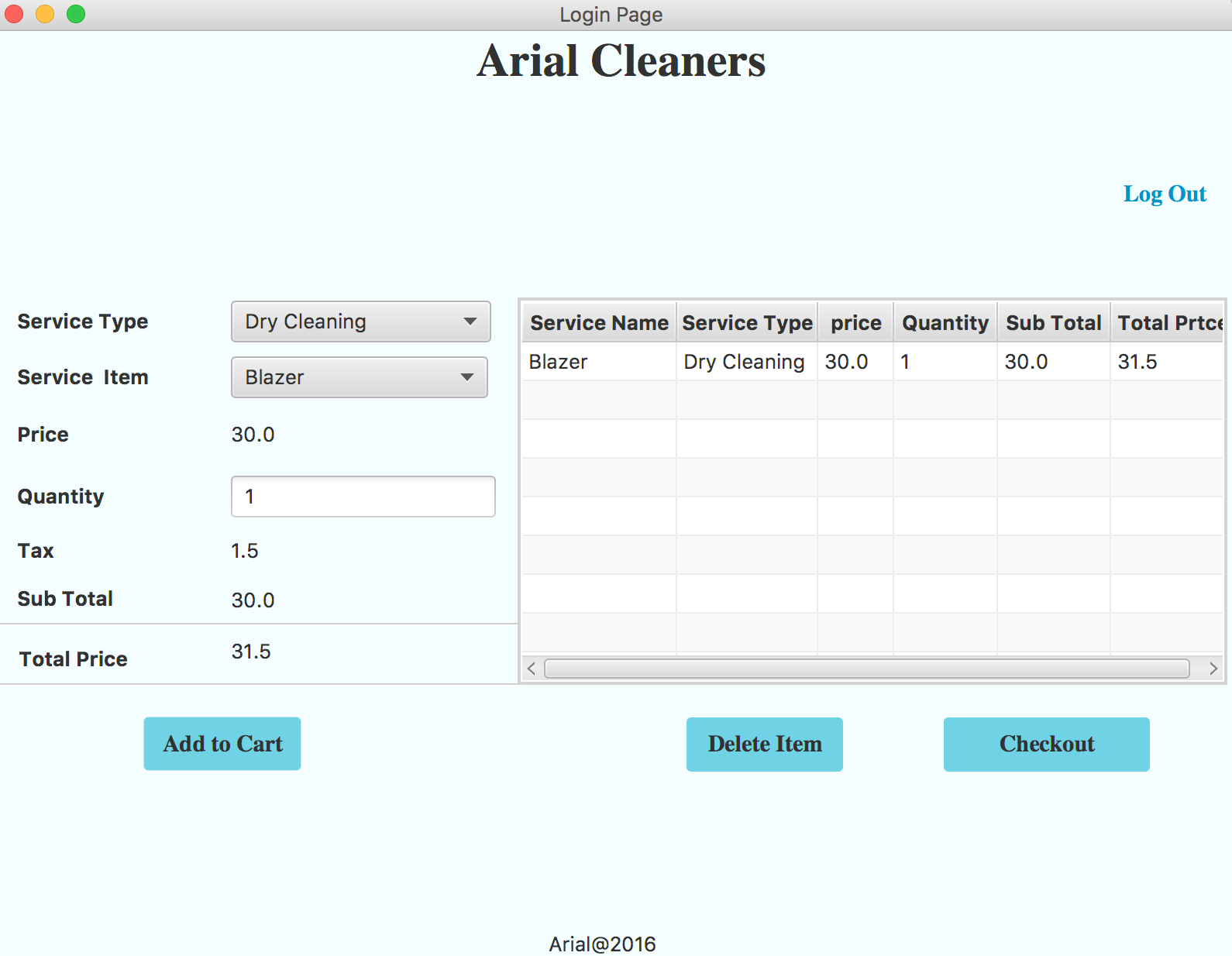


1. Once the customer logs in, He will be prompted with the service cart view. He can then select the service type for ex: dry-cleaning or washing, select the service item for example a blazer, shirt, skirt etc. and put quantity (number of service items) and the price will be automatically prompted with the tax and final amount will be shown at the end. All the service items selected with their respective price will be shown as a list on the same page.

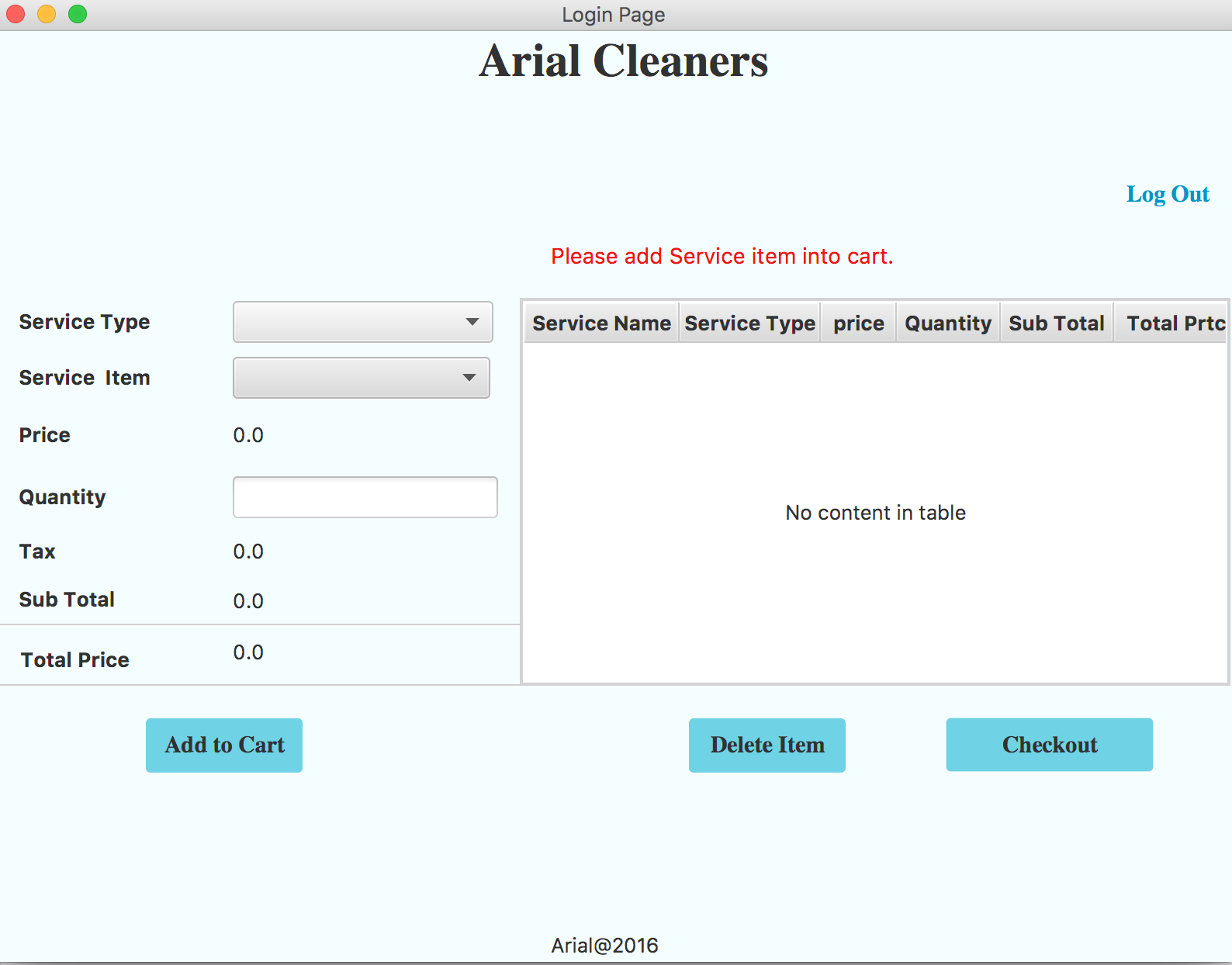
Here, the customers are also given the privileges of adding or deleting a service item by clicking on the ADD and DELETE buttons on this page. Finally, the customer can click on CHECKOUT button to proceed.

Once the Service type and service item is selected and the customer clicks on checkout, the customer will be taken to the confirmation page, where it prompts that the order has been successfully placed.

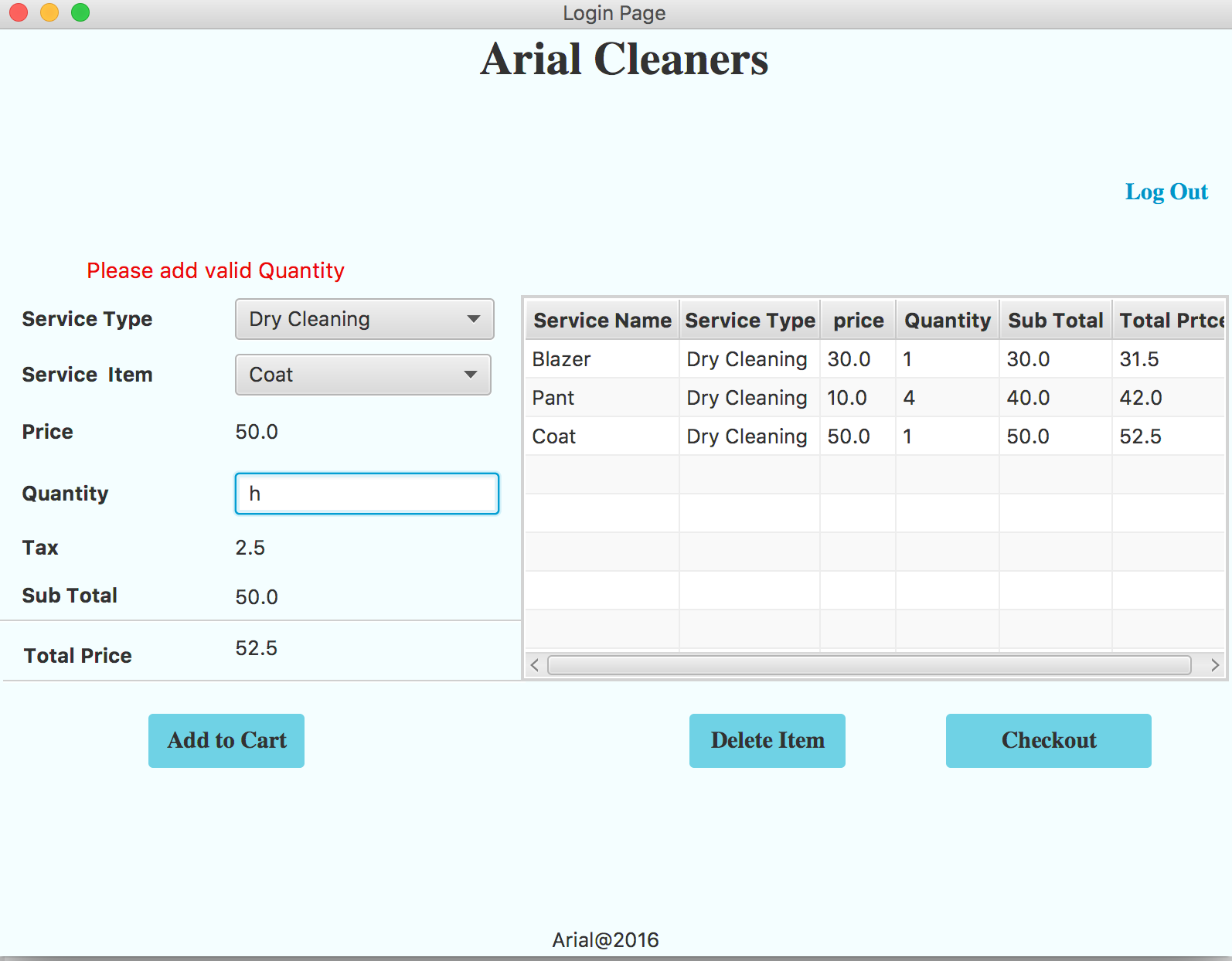
**Service page where only one item has been selected:**



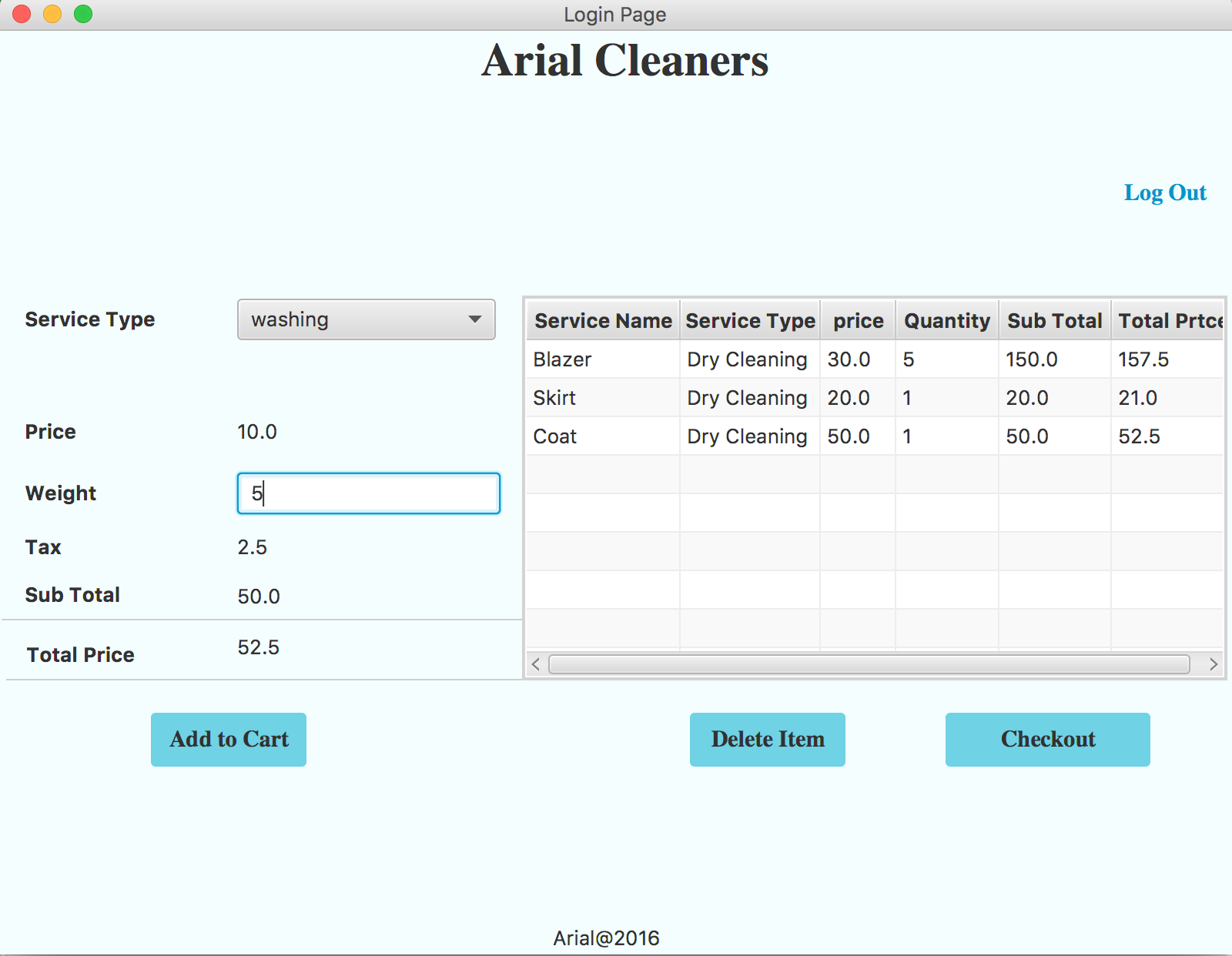
**Error message shown if the cart is empty:**



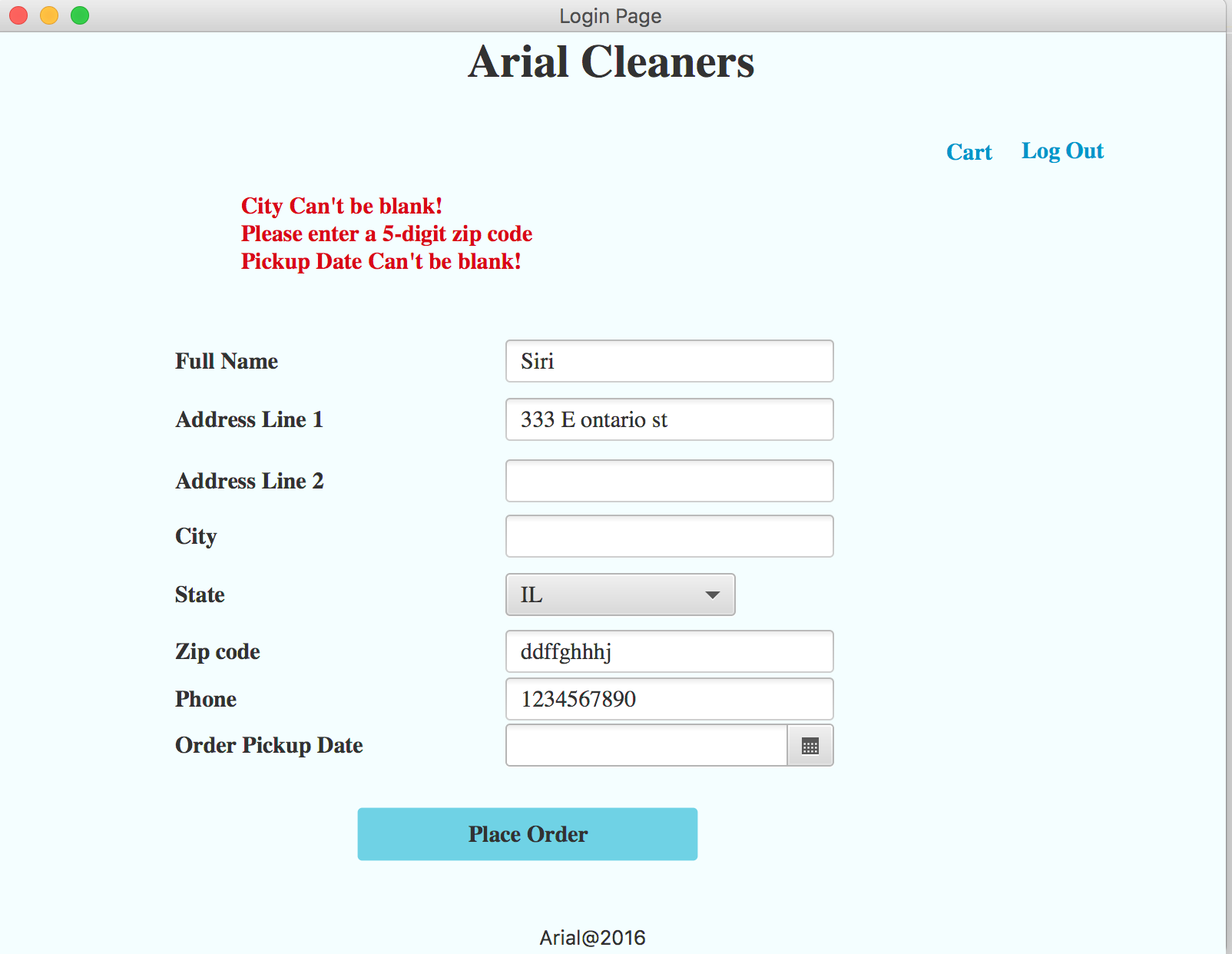
**Validating the quantity area(only valid numbers are expected):**



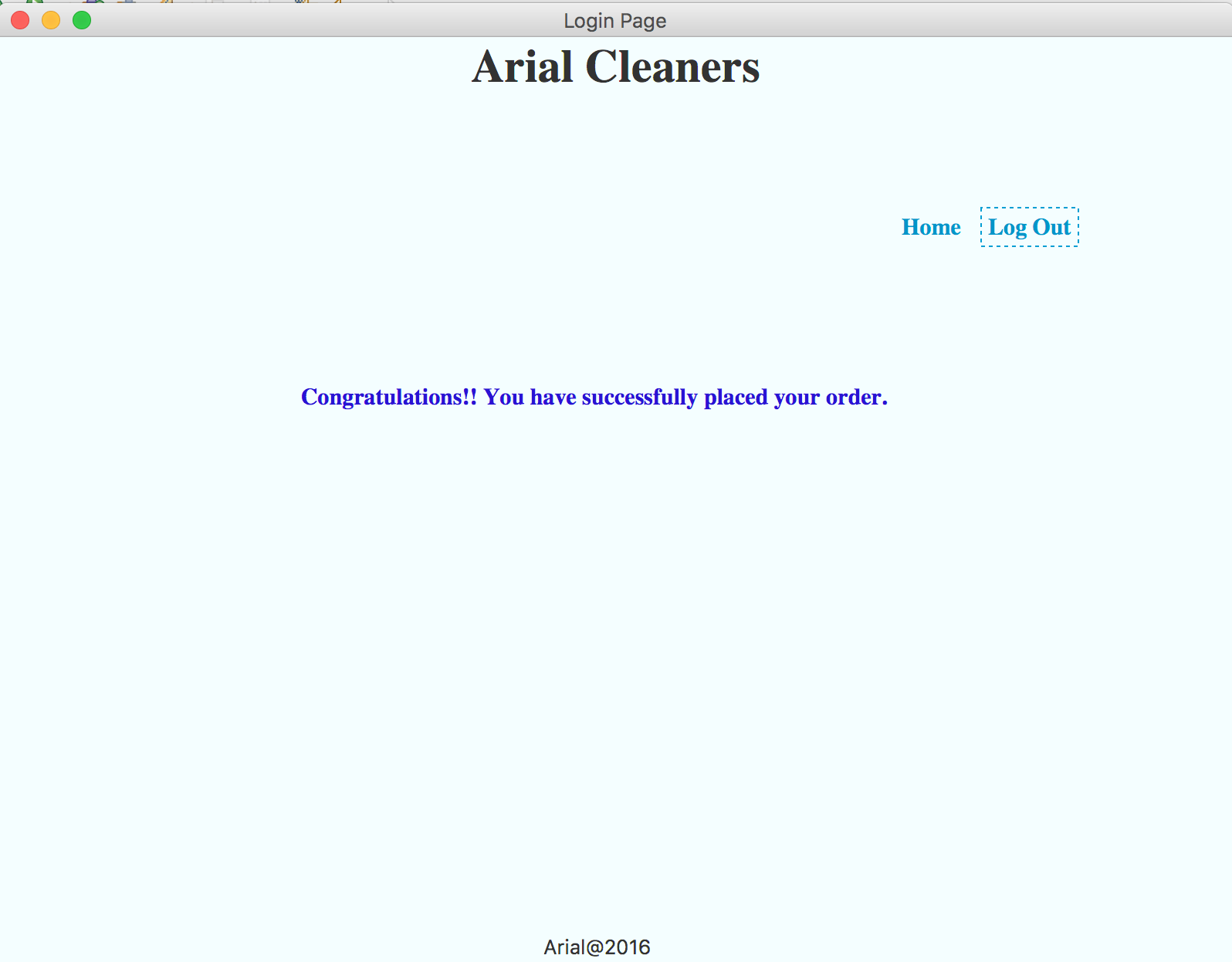
**Adding the weight if the customer selects service type as Washing:**



**Address validation:**

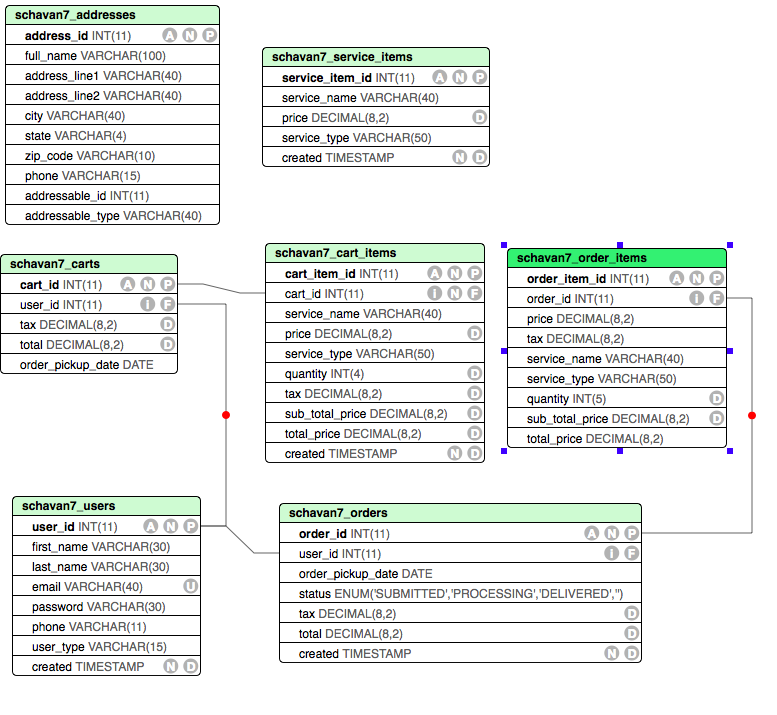


**Order Confirmation page:**



**Java concepts used in the project:**

**ERD:**

****

**Inheritance:**

1. User is a superclass where admin\_user and cutomer\_user extends user class.
2. ItemsModel consists of serviceType, serviceName, price(Attributes) and cartItem, serviceItem and orderItem extends itemsModel class.
3. Dao.java is a superclass for all the Dao classes in the project.

**Model**: Written separate Dao classes for each model classes to perform CURD operations. Following are the Models:

1. User
2. AdminUser
3. Customer
4. Address
5. Item
6. ServiceItem
7. CartItem
8. OrderItem
9. Cart
10. Order.

**Dao Model:**

1. Doa (Super Class): It has two methods to execute Modify query and fetch query.
2. OrderDao, cartDoa, addressDao, customerDao, OrderitemDao, serviceItemDao,,orderItemDao,cartDao, cartItemDao.

**Controllers:**

1. AdminCustomerController: to show customer information to Admin user role.
2. AdminDashController: DashBoard for Admin
3. AdminOrdersController: show orders placed by customer(CURD) to Admin.
4. AdminServiceItemController: Admin performs CURD operation on serviceItem.
5. AuthenticationController: Handles user Login logout and user Registration part.
6. CheckoutController: To allow cart checkout and take order pickup address from user
7. ServiceCartController: User can create serviceItem cart from services selected.
8. SessionManager:
9. Main

**View:** All the GUI code will be executed from the view model.