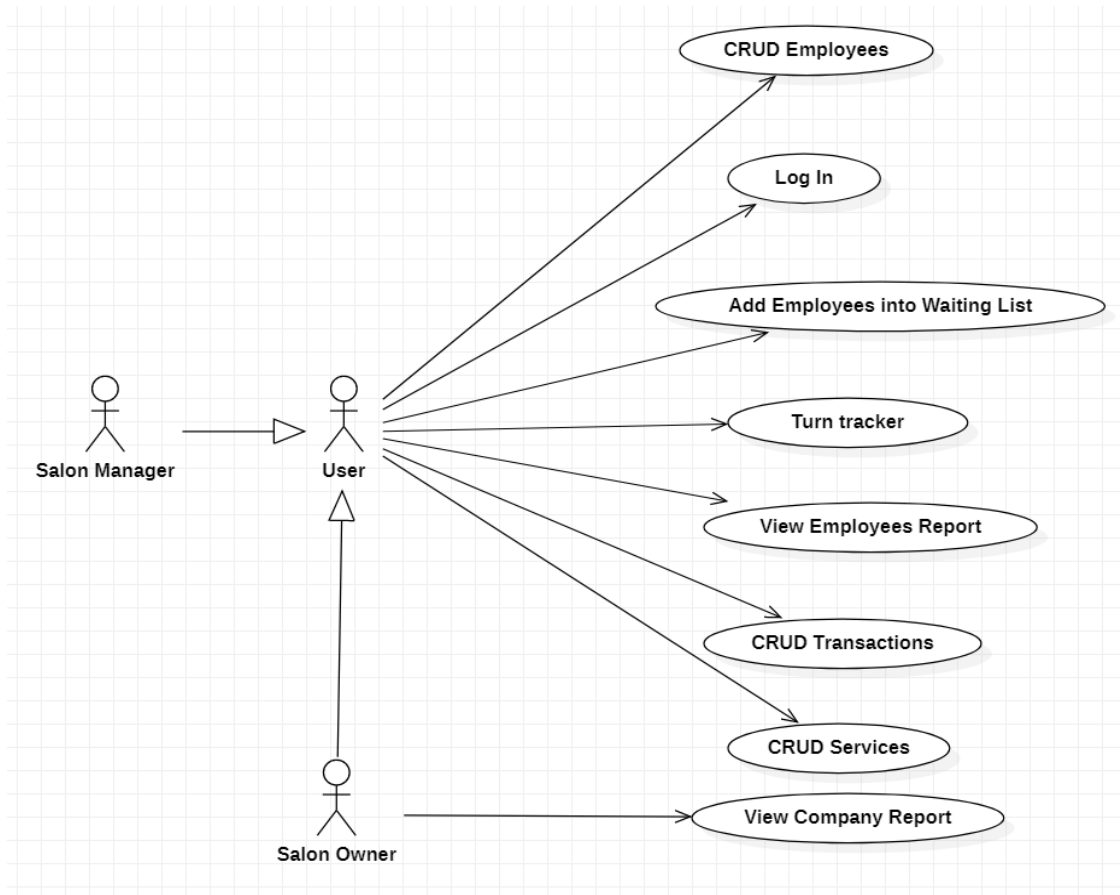


Student Name: *Phuong Nguyen*

Beauty Salon Management System: Use Case Description



| | |
|-----------------------------------|---|
| Use case Number | 1 |
| Name | Log in |
| Brief description | This use case allows user to login |
| Actors | User (salon Manager, salon Owner) |
| Preconditions | Users already have an account |
| User Action | System Response |
| User enters username and password | Verify if the username and password is correct. |
| Post-Conditions | Authorized users can access the system based on their roles |
| Business Rules | Only registered members can login to the system |
| | |
| Use case Number | 2 |

| | | |
|-------------------------------|---|---|
| Name | Employee CRUD | |
| Brief description | This use case allows the owner and manager to create profiles for employees | |
| Actors | Owner, manager | |
| Preconditions | The user must logged in to the system | |
| Flows of Events: | | |
| 1. Basic Flows | | |
| 1.1.0 Create Employee Profile | | |
| Step | User Actions | System Actions |
| 1 | The user clicks on Add New button | The system displays the employee profile form with the fields for firstname, lastname, phone number |
| 2 | The user fills out the form and requests the system to save the details | The system verifies that there’s no other profile in the database with the same name and saves the employee and returns the success message on success or a fail message in case of failure. In case another profile exists with the employee name, the system returns the message indicating a duplicate entry exists. |
| Postconditions | The employee profile is persisted in the system | |
| Business Rules | No duplicate employee profiles. A unique profile is identified by employee's name | |
| 1.1.1 Read/view Employee List | | |
| Step | User Actions | System Actions |
| 1 | The user selects to view a list of employees | The system returns a list of all employees |

| 1.1.2 Update Employee Profile | | |
|-------------------------------|---|---|
| Step | User Actions | System Action |
| 1 | The user selects to view a list of employees | The system returns a list of all employees |
| 2 | The user selects the profile of the employee they want to update | The system displays an editable employee profile form pre-populated with the employee profile details |
| 3 | The user updates the fields they want to update and requests system to save the new details | The system updates the record and returns the success message or a fail message on exception. |
| Postconditions | The staff profile will be updated | |
| Business Rule | The ID field should be unwritable. | |
| 1.1.3 Delete Employee Profile | | |
| Step | User Actions | System Actions |
| 1 | The user selects to view a list of employees | The system returns a list of employees |
| 2 | The user selects to delete an employee from a list of employees | The system displays a confirmation dialogue window |
| 3 | The user selects OK on the confirmation dialog window to confirm deleting the profile | The system deletes the profile. The system returns message a success message. |
| Postconditions | The employee profile will be deleted | |
| | | |
| Use case Number | 3 | |
| Name | Service CRUD | |
| Brief description | This use case allows the owner to create a new service | |

| | | |
|------------------------------|---|---|
| Actors | Owner, manager | |
| Preconditions | The user must logged in to the system | |
| Flows of Events: | | |
| 1. Basic Flows | | |
| 1.1.0 Create a new service | | |
| Step | User Actions | System Actions |
| 1 | The user clicks on Add New button | The system displays the service form with the fields for service name, service price and the note |
| 2 | The user fills out the form and requests the system to save the details | The system verifies that there’s no other service in the database with the same service name and saves the service and returns the success message on success or a fail message in case of failure. In case another service exists with the service name, the system returns the message indicating a duplicate entry exists. |
| Postconditions | The new service is persisted in the system | |
| Business Rules | No duplicate service. A unique service is identified by service name | |
| 1.1.1 Read/view Service List | | |
| Step | User Actions | System Actions |
| 1 | The user selects to view a list of services | The system returns a list of all services |
| 1.1.2 Update Service | | |
| Step | User Actions | System Action |

| | | |
|-----------------------|---|---|
| 1 | The user selects to view a list of services | The system returns a list of all services |
| 2 | The user selects the services they want to update | The system displays an editable service form pre-populated with the service details |
| 3 | The user updates the fields they want to update and requests system to save the new details | The system updates the record and returns the success message or a fail message on exception. |
| Postconditions | The service will be updated | |
| Business Rule | The ID field should be unwritable. | |

1.1.3 Delete service

| Step | User Actions | System Actions |
|-----------------------|---|---|
| 1 | The user selects to view a list of services | The system returns a list of services |
| 2 | The user selects to delete a service from a list of services | The system displays a confirmation dialogue window |
| 3 | The user selects OK on the confirmation dialog window to confirm deleting the service | The system deletes the service. The system returns message a success message. |
| Postconditions | The service will be deleted | |

| | |
|--------------------------|--|
| Use case Number | 4 |
| Name | Transaction CRUD |
| Brief description | This use case allows the owner and manager to create a new transaction |
| Actors | Owner, manager |

| | | |
|--------------------------------|---|--|
| Preconditions | The user must logged in to the system | |
| Flows of Events: | | |
| 1. Basic Flows | | |
| 1.1.0 Create a new transaction | | |
| Step | User Actions | System Actions |
| 1 | The user clicks on Add New button | The system displays the transaction form with the fields for service name, payment method, tip |
| 2 | The user fills out the form and requests the system to save the details | The system returns the success message on success or a fail message in case of failure. |
| Postconditions | The new transaction is persisted in the system | |
| Business Rules | Service name, Payment Method are required field. Tip field is optional. | |
| 1.1.1 Read/view Service List | | |
| Step | User Actions | System Actions |
| 1 | The user selects to view a list of transactions | The system returns a list of all transactions |
| 1.1.2 Update Transaction | | |
| Step | User Actions | System Action |
| 1 | The user selects to view a list of transactions | The system returns a list of all transactions |
| 2 | The user selects the transactions they want to update | The system displays an editable transaction form pre-populated with the transaction details |

| | | |
|-----------------------|---|---|
| 3 | The user updates the fields they want to update and requests system to save the new details | The system updates the record and returns the success message or a fail message on exception. |
| Postconditions | The transaction will be updated | |
| Business Rule | The ID field should be unwritable. | |

1.1.3 Delete Transaction

| Step | User Actions | System Actions |
|-----------------------|---|---|
| 1 | The user selects to view a list of transactions | The system returns a list of transactions |
| 2 | The user selects to delete a transaction from a list of transactions | The system displays a confirmation dialogue window |
| 3 | The user selects OK on the confirmation dialog window to confirm deleting the transaction | The system deletes the transaction. The system returns message a success message. |
| Postconditions | The transaction will be deleted | |

| | |
|--------------------------|--|
| Use case Number | 5 |
| Name | View Employees Report |
| Brief description | This use case allows user to view employee report. |
| Actors | User (salon Manager, salon Owner) |
| Preconditions | The user must logged in to the system |
| User Action | System Response |

| | |
|---|---|
| The user selects to view a report they want | The system returns the employee report page |
| The user selects from date to date and hit submit | The system returns the employee report object as a string with employee name, and total they made from date to date |
| Post-Conditions | The employee report will be showed |
| Business Rules | From date to date picker is optional, by default it will show the current date. |

| | |
|---|--|
| Use case Number | 6 |
| Name | View Company Report |
| Brief description | This use case allows the owner to view company report. |
| Actors | salon owner |
| Preconditions | The owner must logged in to the system |
| User Action | System Response |
| The owner selects to view a report they want | The system returns the company report page |
| The owner selects from date to date, payment type from dropdown menu and hit submit | The system returns the company report object as a string with service ID, service date, payment type, amount and total amount from date to date |
| Post-Conditions | The company report will be showed |
| Business Rules | From date to date picker is optional, by default it will show the current date. Payment dropdown menu is optional by default it will show all of the transactions. |

| | |
|---|--|
| | |
| Use case Number | 7 |
| Name | Add employees into Waiting List |
| Brief description | This use case allows the user to add employees into Waiting List |
| Actors | user (owner, manager) |
| Preconditions | The user must logged in to the system |
| User Action | System Response |
| The user clicks the IN button to add the employee into the Waiting List | The system returns the updated Waiting List by order first in first out |
| Post-Conditions | The employee will be added into the Waiting List by order first in first out |

| | |
|--------------------------|--|
| | |
| Use case Number | 8 |
| Name | Turn Tracker |
| Brief description | This usecase allows the user to manage employee's turn in the waiting list |
| Actors | user (owner, manager) |
| Preconditions | The user must logged in to the system |
| User Action | System Response |

| | |
|---|---|
| The user clicks on the employee's name (button) to manage the employee's turn in the waiting list | The system returns the updated Waiting List which the employee's name was clicked will be moved to very bottom of the list (first in first out) |
| Post-Conditions | The updated Waiting list will be showed |