Randy Nguyen, Andrew Driscoll, Gareth Campbell

CSC 380

**Restaurant Management System**

* (1) Project description: a narrative in English, written by a potential client or user of your software product
* (2) a list of system requirements, together with priorities, as described in class
* (3) the same list in the form of user stories, together with a size estimate, as described in class

**1. Project Description:**

Restaurant Management System key features

* Login Screen w/ Password/Time Management
* Ordering Service/Food Menu Buttons
* Inventory Management
* Tables/Customers Management

A restaurant is in many ways a machine with multiple moving moving parts. All must work simultaneously and do its individual function for the machine to work properly. Often times this process can become hectic and even to this day, many non-franchised restaurants have a difficult time with tasks such as staying organized, keeping track of inventory, and managing employee hours. The development of a software geared towards smaller restaurants would help owners, managers, and all parties involved operate a successful restaurant.

The software will be a restaurant management system. It should allow for the tracking and processing of multiple restaurant functions such as allowing an employee to sign in to the terminal and clocking in for the day. The software will then track and maintain a log of the time from sin in until sign out of all employees. Upon clocking in, the employee should be assigned a section of tables to work and be able to place orders for the given table It should allow for the tracking of inventory so that when an employee places an order the inventory is updated automatically to represent the usage of materials. The software should also allow for the generation of a bill for each table when requested. Upon command, the software should generate a receipt for the current transaction. Finally, the employee should be able to clock out at the end of the workday. The employee should be able to sign out of the terminal at any time allowing someone else to sign in while maintaining their table information.

All of the software should be used through a graphic user interface that allows for using a mouse for input. The menus displayed should appropriately separate the functions by category such as sign in/sign out, place order, and generate the bill.

**2. System Requirements:**

|  |  |  |
| --- | --- | --- |
| Identifier: | Priority (1(Low) – 5(High): | Requirement: |
| REQ1 | 5 | The system shall keep itself password locked, unless unlocked by an authorized user. An authorized login made up of a username and key-code will bring the system to the next interactive page. |
| REQ2 | 3 | The system shall return to its locked state when commanded by pressing a designated button. |
| REQ3 | 5 | The system shall have a home screen to navigate through different pages/screens. |
| REQ4 | 3 | The system should shall maintain a log of employee hours from the time of login until the time of logout. |
| REQ5 | 4 | The system shall maintain a log of all available inventory and alert the user when an item’s inventory is 0. An authorized user should be able to restock empty inventory. |
| REQ6 | 3 | The system should allow users to add, manage, and remove customers from a table. |
| REQ7 | 4 | The system should allow the user to select and deduct customer orders from the inventory when commanded by pressing a designated menu item button for a specific table. |
| REQ8 | 3 | The system should include a checkout function for each designated table and generate the table bill. |
|  |  |  |

**SYSTEM REQUIREMENTS (CONTINUED)**

|  |  |  |
| --- | --- | --- |
| REQ9 | 1 | The system should maintain a log of all bills. |
| REQ10 | 4 | The system should allow an Administrator to update the menu. |
| REQ11 | 3 | The system should allow an Administrator to create new employee accounts. |
| REQ12 | 2 | The system should allow an Administrator to delete existing employee accounts. |
| REQ13 | 4 | The system shall designate between employees and administrators and grant permissions based upon which type of account is currently signed in. |

**3. User Stories:**

|  |  |  |
| --- | --- | --- |
| Identifier: | User Story: | Size: |
| ST-1 | As an authorized user I can securely log on the system. | 5 points |
| ST-2 | As an authorized user, I can log off the system when commanded. | 3 points |
| ST-3 | As an authorized user, I can easily access the system page I need. | 9 points |
| ST-4 | As an authorized user, I can log and track how long I’ve been working. | 6 points |
| ST-5 | As an authorized user, I can track how much inventory is available. | 7 points |
| ST-6 | As an authorized user, I can view and manage customer orders and transactions. | 9 points |
| ST-7 | As an authorized user, I can select from menu items to take customer orders. | 5 points |
|  |  |  |

**USER STORIES (CONTINUED)**

|  |  |  |
| --- | --- | --- |
| ST-8 | As an authorized user, I can generate the bill for a customer. | 6 points |
| ST-9 | As an authorized user, I can access a sales history detailing previous sales totals. | 5 points |
| ST-10 | As an authorized administrator, I can add new items to the menu. | 4 points |
| ST-12 | As an authorized administrator, I can create new employee accounts. | 4 points |
| ST-11 | As an authorized administrator, I can delete existing employee accounts. | 4 points |
| ST-12 | As an authorized user, I will be presented actions depending on what  level of privileges I have. | 6 points |

**4. Use Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-1:** | | | | **LOGIN TO SYSTEM** |
| **Related Requirements:** | | | REQ1, REQ2, and REQ3 stated in the System Requirements Table on page 2. | |
| **Initiating Actor:** | | | Any user who opens the system/ Authorized user / Administrator | |
| **Actor’s Goal:** | | | To securely access the entire restaurant management system. | |
| **Participating Actors:** | | | LoginScreen, AccountFolder, and HomeScreen | |
| **Preconditions:** | | | The set of authorized login accounts (AccountFolder) stored in the system database is not empty.  The system displays an area where the user can enter username and pin. | |
| **Postconditions:** | | | The system stays logged in and will not require another login until logged out. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **User** turns on/opens the system. | | |
| ← | 2. | **System** displays the **LoginScreen**. | | |
| ← | 3. | **System** prompts the **user** to enter a username and pin. | | |
| → | 4. | **Authorized user** enters a correct login that matches with an account in **AccountFolder**. | | |
| ← | 5. | **System** notifies **Authorized user** of successful login and displays **HomeScreen.** | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-2:** | | | | **LOGOUT OF SYSTEM** |
| **Related Requirements:** | | | REQ1, REQ2, and REQ3 stated in the System Requirements Table on page 2. | |
| **Initiating Actor:** | | | Any user of the system. | |
| **Actor’s Goal:** | | | To secure the system after use. | |
| **Participating Actors:** | | | LogoutButton, ConfirmationMessage, and LoginScreen. | |
| **Preconditions:** | | | None. | |
| **Postconditions:** | | | The system displays the LoginScreen. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **User** selects the **LogoutButton.** | | |
| ← | 2. | **System** displays **ConfirmationMessage**- a message to verify that the user wishes to logout. | | |
| → | 3. | **User** selects “yes”. | | |
| ← | 4. | **System** returns to **LoginScreen.** | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-3:** | | | | **PLACE AN ORDER/ ADD TABLE** |
| **Related Requirements:** | | | REQ6 and REQ7 stated in the System Requirements Table on page 2. | |
| **Initiating Actor:** | | | Employee | |
| **Actor’s Goal:** | | | To place an order for a customer and manage where they are seated, while updating the inventory. | |
| **Participating Actors:** | | | Menu, MenuButtons, Table, AdministratorMenu, Timer, and Inventory | |
| **Preconditions:** | | | The selected table is not preoccupied by another customer.  Inventory is not empty. | |
| **Postconditions:** | | | Timer is started after order is placed and stops when food is delivered.  The table remains occupied until customers have paid bill and checked out. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **Employee** arrives at the terminal and selects a **table.** | | |
| → | 2. | **Employee** selects **menuButtons** on **menu** to place an order for selected **table.** | | |
| ← | 3. | **System** deducts items selected on **menu** from **inventory**. | | |
| ← | 4. | **System** starts **timer** when order is placed. | | |
| → | 5. | **Employee** delivers the food and marks the **table** as delivered. | | |
| ← | 6. | **System** stops the **timer**. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-4:** | | | | **GET BILL/CLEAR TABLE** |
| **Related Requirements:** | | | REQ1, REQ5, REQ6, REQ8, and REQ9 stated in the System Requirements Table on page 2-3 | |
| **Initiating Actor:** | | | Employee | |
| **Actor’s Goal:** | | | To display the bill, clear the table of customer information, and store the bill information. | |
| **Participating Actors:** | | | Table, Authorized User | |
| **Preconditions:** | | | * The table must have customer order information. * The Employee must be an authorized user. * The Customer must be finished ordering. | |
| **Postconditions:** | | | The table information will be stored and cleared. The bill information will be produced. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **Employee** unlocks the system. | | |
| → | 2. | **Employee** selects the “Checkout” option from menu. | | |
| ← | 3. | **System** signals to the **Table** to (a) CalculateBill (b) ReturnBill (c) StoreBill (d) AddEndTime | | |
| ← | 4. | **System** signals to **Employee** that **table** is closed. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-5:** | | | | **EMPLOYEE SIGN IN** |
| **Related Requirements:** | | | REQ1, REQ3, and REQ4 stated in the System Requirements Table on page 2-3 | |
| **Initiating Actor:** | | | Employee | |
| **Actor’s Goal:** | | | Employee will have sign in time logged into employee tracking system. | |
| **Participating Actors:** | | | Employee, AuthorizedUser | |
| **Preconditions:** | | | * Employee is an authorized user. * Employee is not currently signed in for the day. | |
| **Postconditions:** | | | Employee’s sign-in time will be recorded from the system clock into the employee object. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **Employee** will log-in into system. | | |
|  | 2. | include::*AuthenticateUser* | | |
| ← | 3. | **System** (a) Retrieves time stamp from system clock. (b) Enters time stamp into **Employee** object. | | |
| ← | 4. | **System** signals to the **Employee** to output confirmation message. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-6:** | | | | **EMPLOYEE SIGN OUT** |
| **Related Requirements:** | | | REQ1, REQ2, REQ3, and REQ4 stated in the System Requirements Table on page 2-3 | |
| **Initiating Actor:** | | | Employee | |
| **Actor’s Goal:** | | | Employee will have sign out time logged into employee tracking system. | |
| **Participating Actors:** | | | Employee, AuthorizedUser | |
| **Preconditions:** | | | * Employee is an authorized user. * Employee is currently signed in for the day. * Employee is currently not signed out for the day. | |
| **Postconditions:** | | | Employee’s sign-out time will be recorded from the system clock into the employee object. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **Employee** will log-in into system. | | |
|  | 2. | include::*AuthenticateUser* (UC-1) | | |
| ← | 3. | **System** (a) Retrieves time stamp from system clock. (b) Enters time stamp into **Employee** object. | | |
| ← | 4. | **System** signals to the **Employee** to output confirmation message. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-7:** | | | | **ADMIN ONLY FUNCTION: ADD USER** |
| **Related Requirements:** | | | REQ1, REQ3, REQ11 and REQ13 stated in the System Requirements Table on page 2-3. | |
| **Initiating Actor:** | | | Administrator | |
| **Actor’s Goal:** | | | To create a new employee account | |
| **Participating Actors:** | | | AdministratorMenu and AccountFolder | |
| **Preconditions:** | | | • There exist a pin that allows for the Administrator to sign in.  • The system displays prompts for required info and allows for its input. | |
| **Postconditions:** | | | A new user account has been created and stored in the account folder. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **Administrator** arrives at terminal and signs in. | | |
| → | 2. | **Administrator** selects the option to add a new employee in **AdministratorMenu**. | | |
| → | 3. | **Administrator** enters prompted for employee information. | | |
| ← | 4. | The new **Employee** account is added to the **AccountFolder** and a confirmation message is displayed. | | |
| → | 5. | The **Administrator** selects next action to complete from **AdministratorMenu**. | | |

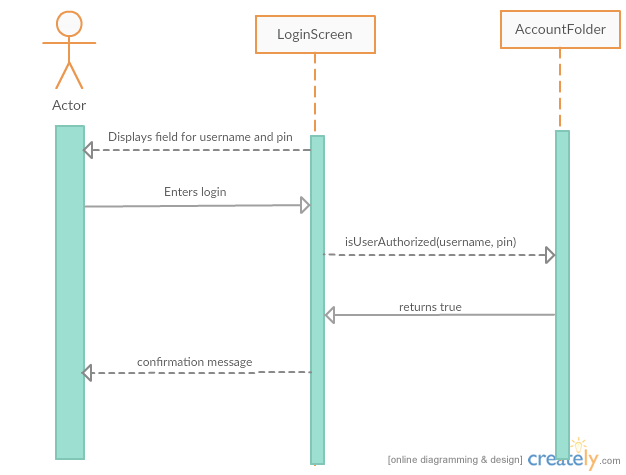
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-8:** | | | | **ADMIN ONLY FUNCTION: DELETE USER** |
| **Related Requirements:** | | | REQ1, REQ3, REQ12 and REQ13 stated in the System Requirements Table on page 2-3. | |
| **Initiating Actor:** | | | Administrator | |
| **Actor’s Goal:** | | | To delete employee account | |
| **Participating Actors:** | | | AdministratorMenu and AccountFolder | |
| **Preconditions:** | | | • The user account specified exists  • The system displays prompts for required info and allows for its input. | |
| **Postconditions:** | | | The account has been removed from the account file. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **Administrator** arrives at terminal and signs in. | | |
| → | 2. | **Administrator** selects the option to delete an existing employee in **AdministratorMenu**. | | |
| → | 3. | **Administrator** enters name of employee account to delete. | | |
| ← | 4. | The **Employee** account is deleted from the **AccountFolder** and a confirmation message is displayed. | | |
| → | 5. | The **Administrator** selects next action to complete from **AdministratorMenu**. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-9:** | | | | **ADMIN ONLY FUNCTION: UPDATE MENU** |
| **Related Requirements:** | | | REQ1, REQ3, REQ10 and REQ13 stated in the System Requirements Table on page 2-3. | |
| **Initiating Actor:** | | | Administrator | |
| **Actor’s Goal:** | | | To add a new item to the menu | |
| **Participating Actors:** | | | AdministratorMenu and Menu | |
| **Preconditions:** | | | • The menu has been loaded into the AdministratorMenu successfully  • The system displays prompts for required info and allows for its input.  • The menu item to be added to the Menu does not already exist in the Menu. | |
| **Postconditions:** | | | The new item has been added to the Menu. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **Administrator** arrives at terminal and signs in. | | |
| → | 2. | **Administrator** selects the option to add a new item to the **Menu**. | | |
| → | 3. | **Administrator** enters the prompted for item information | | |
| ← | 4. | The new menu item is added to **Menu** and a confirmation message is displayed. | | |
| → | 5. | The **Administrator** selects next action to complete from **AdministratorMenu**. | | |

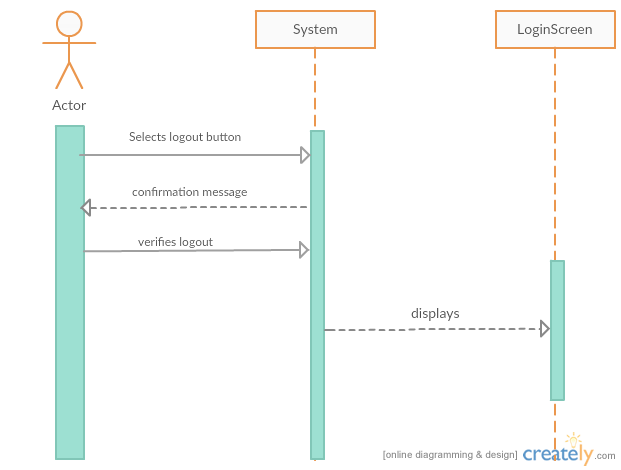
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case UC-10:** | | | | **ADMIN ONLY FUNCTION: UPDATE INVENTORY** |
| **Related Requirements:** | | | REQ1, REQ4, REQ10, and REQ13 stated in the System Requirements Table on page 2-3. | |
| **Initiating Actor:** | | | Administrator | |
| **Actor’s Goal:** | | | To update inventory numbers. | |
| **Participating Actors:** | | | AdministratorMenu and Menu | |
| **Preconditions:** | | | • The menu has been loaded into the AdministratorMenu successfully.  • The system displays prompts for required info and allows for its input.  • The inventory to be updated must be for an already existing menu item in the Menu. | |
| **Postconditions:** | | | The inventory amount for the selected item is updated. | |
| **Flow of Events for Main Success Scenario:** | | | | |
| → | 1. | **Administrator** arrives at terminal and signs in. | | |
| ← | 2. | **Administrator** selects the option to update the inventory of a menu item. | | |
| → | 3. | **Administrator** enters name of the menu item to update inventory for. | | |
| → | 4. | **Administrator** enters the amount to adjust the current inventory number by. | | |
| ← | 5. | The updated inventory amount is inserted into the **Menu** and a confirmation message is displayed. | | |
| → | 6. | The **Administrator** selects next action to complete from **AdministratorMenu**. | | |

**5. System Sequence Diagrams**

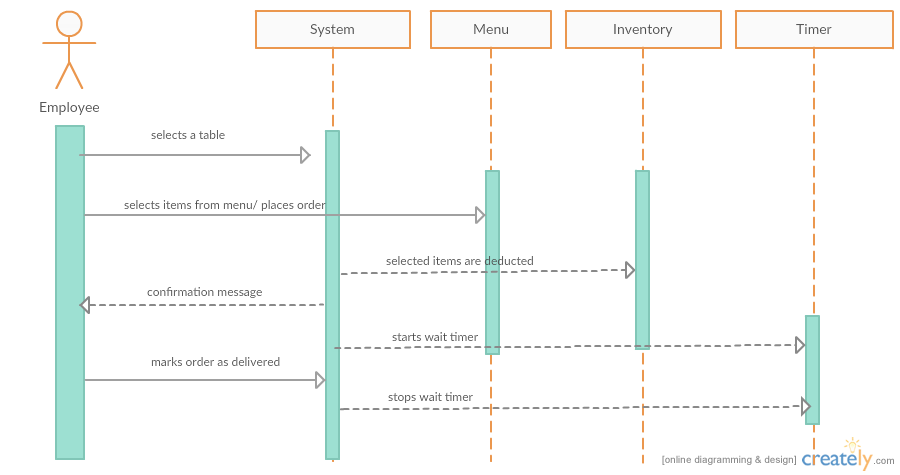
SSD for Use Case 1: Login to System



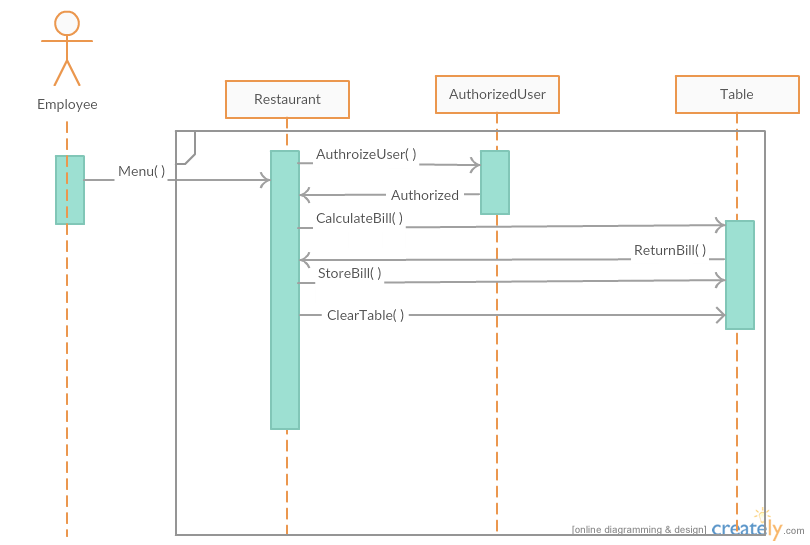
SSD for Use Case 2: Logout of System



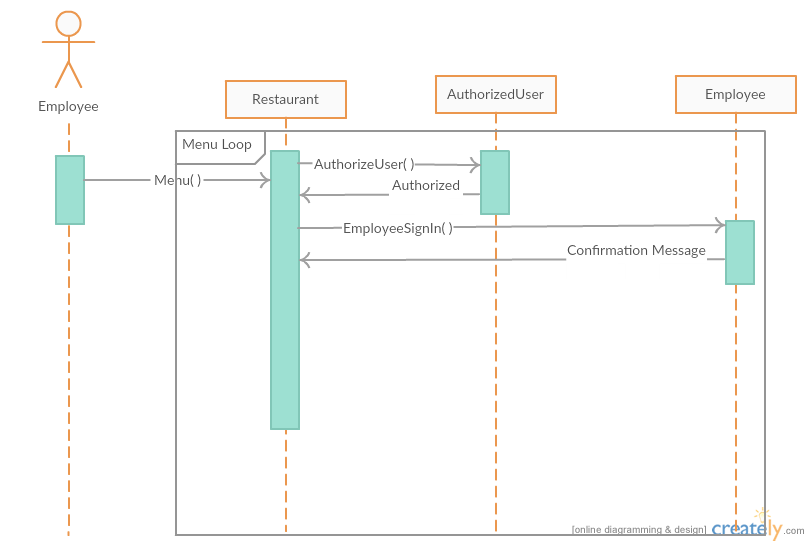
SSD for Use Case 3: Place Order/Add a Table



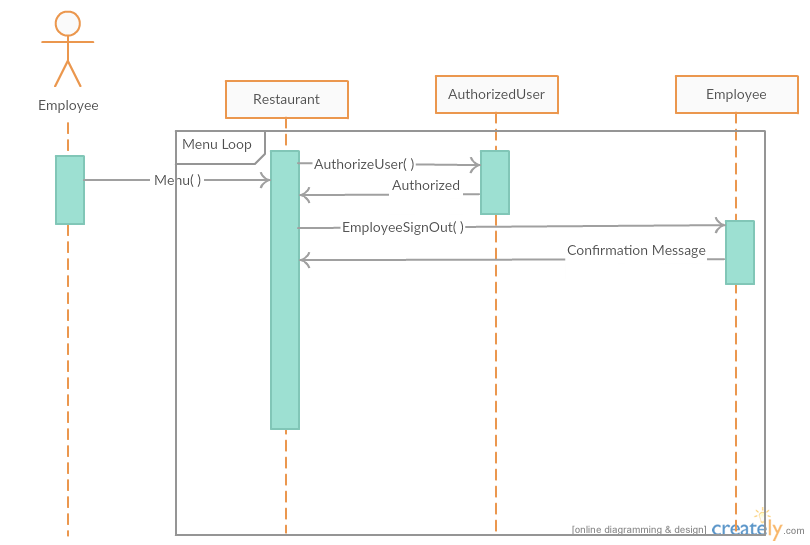
SSD for Use Case 4: Get Bill/ Clear Table



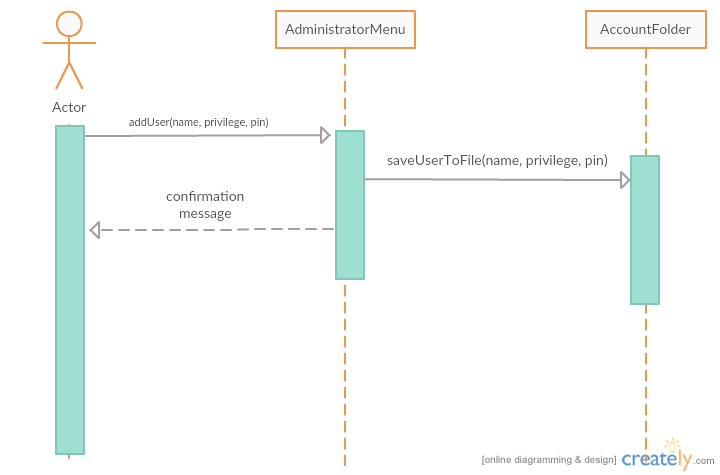
SSD for Use Case 5: Employee Sign In



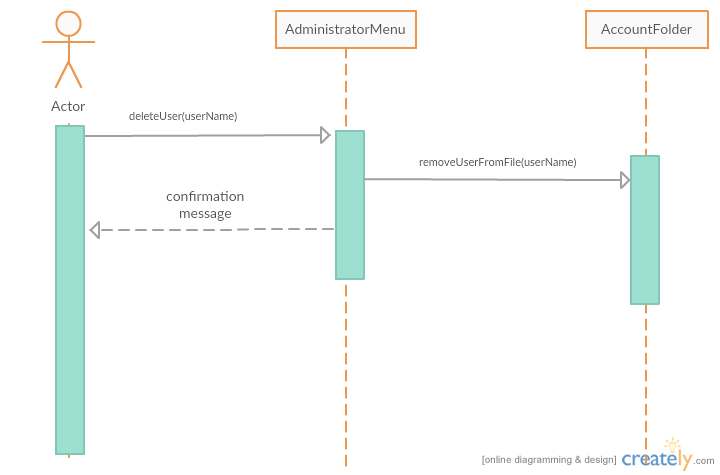
SSD for Use Case 6: Employee Sign Out



SSD for Use Case 7: Admin: Add User



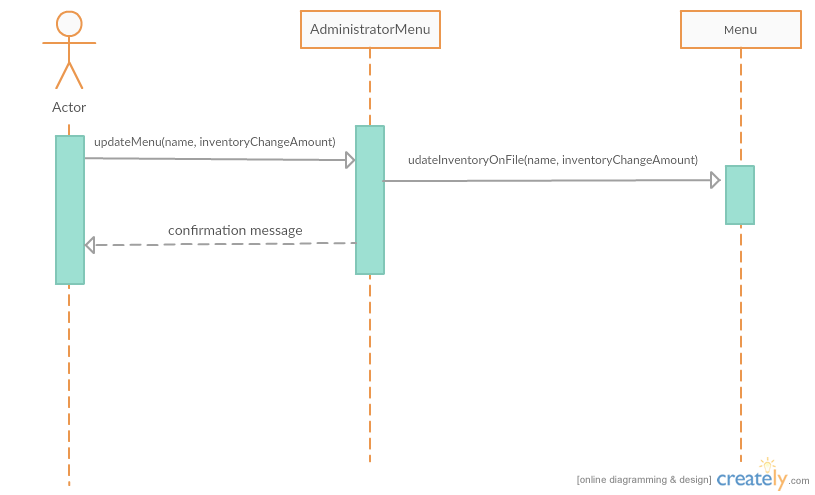
SSD for Use Case 8: Admin: Delete User



SSD for Use Case 9: Admin: Update Menu



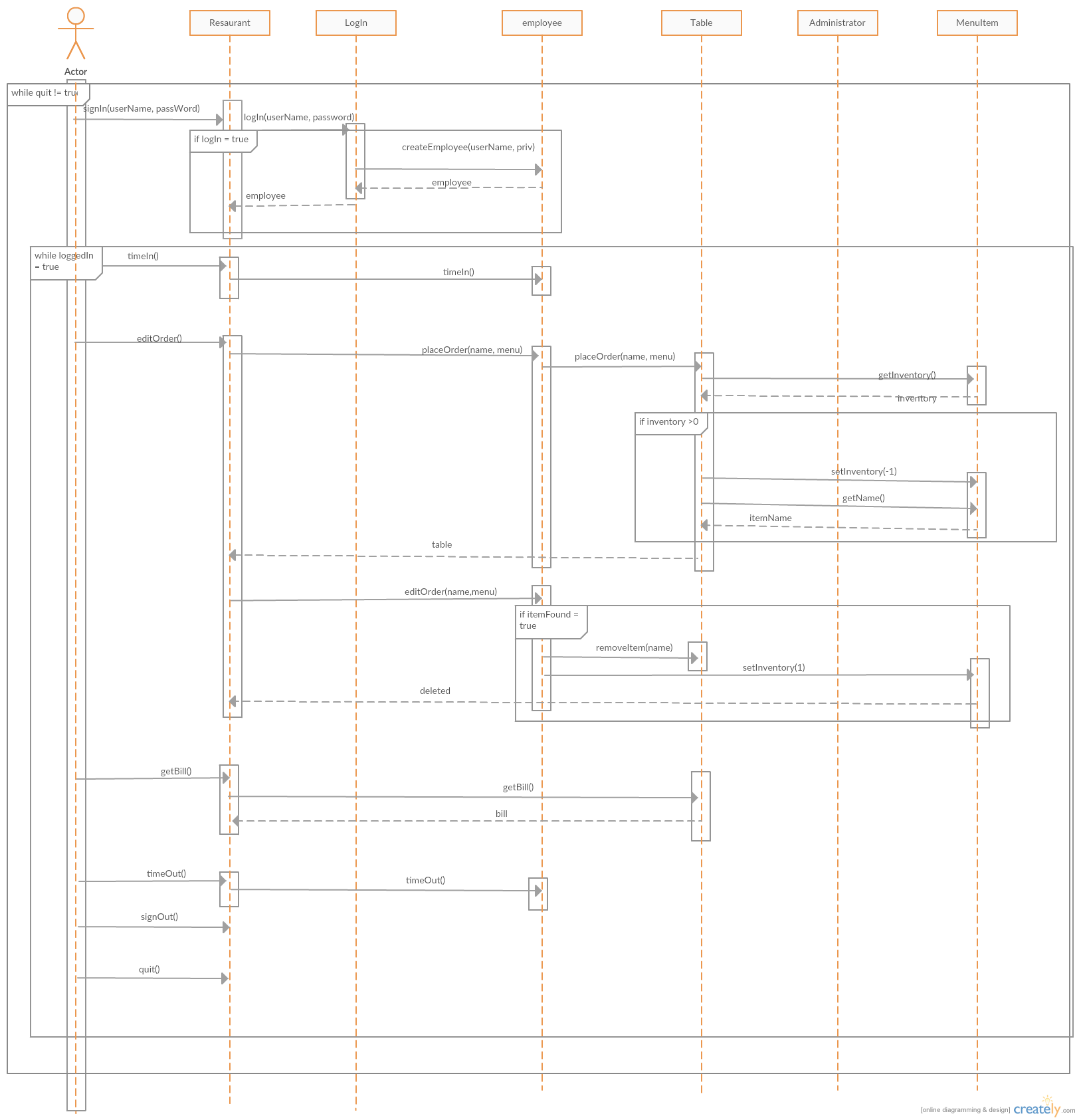
Use Case 10: Admin: Update Inventory



**6. Traceability Matrix**

*Maps system requirements to Use Cases*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req't** | **PW** | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | UC8 | UC9 | UC10 |
| REQ1 | 5 | X | X |  | X | X | X | X | X | X | X |
| REQ2 | 3 | X | X |  |  |  | X |  |  |  |  |
| REQ3 | 5 | X | X |  |  | X | X | X | X | X |  |
| REQ4 | 3 |  |  |  |  | X | X |  |  |  |  |
| REQ5 | 4 |  |  |  | X |  |  |  |  |  | X |
| REQ6 | 3 |  |  | X | X |  |  |  |  |  |  |
| REQ7 | 4 |  |  | X |  |  |  |  |  |  |  |
| REQ8 | 3 |  |  |  | X |  |  |  |  |  |  |
| REQ9 | 1 |  |  |  | X |  |  |  |  |  |  |
| REQ10 | 4 |  |  |  |  |  |  |  |  | X | X |
| REQ11 | 3 |  |  |  |  |  |  | X |  |  |  |
| REQ12 | 2 |  |  |  |  |  |  |  | X |  |  |
| REQ13 | 1 |  |  |  |  |  |  | X | X | X | X |
| MAX PW | | 5 | 5 | 3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| AVG PW | | 4.3 | 4.3 | 3.5 | 3.2 | 4.3 | 4 | 3.5 | 3.3 | 3.8 | 3.5 |

**SYSTEM SEQUENCE DIAGRAM**

**UML**

