## **Use case template**

An adaptation of the standard Cockburn template will be used. The template and examples follow:

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC1 - Register and Authenticate Subscriber | | |
| Primary actor | Subscriber | Secondary actors | Library System |
| Description | Allows a new user to register an account or allows a subscriber to authenticate in order to access the library system .  For each subscriber, the system stores national identification number, name, address, phone number, a unique identification code within the library and a password. | | |
| Trigger | A user attempts to use a library terminal. | | |
| Preconditions | * PRE-1. The terminal must be operational. * PRE-2. The user must have valid personal information. | | |
| Postconditions | * POST-1. If registering, the user is added to the system as a subscriber. * POST-2. The subscriber is authenticated in the system. * POST-3. The subscriber gains access to their account. | | |
| Normal flow | 1. Authenticate 2. The user selects "Login" at the terminal. 3. The subscriber enters their unique identification code within the library and password. 4. The system verifies the credentials. 5. If valid, the system grants access and confirms authentication. | | |
| Alternative flows | 1.1 Register   1. The user selects "Register" at the terminal. 2. The system prompts for name, address, phone number, and a unique identification number. 3. The user provides the required details. 4. The system creates a new subscriber account and generates a unique library ID. 5. The system confirms successful registration. | | |
| Exceptions | **1.0.E1** Incomplete or invalid registration details  If the user provides incomplete or invalid registration details, the system displays an error message and prompts for correction.  **1.0.E1** Incorrect login credentials  If the subscriber enters incorrect login credentials, the system displays an error message and allows another attempt. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC2 - Select and Reserve Books | | |
| Primary actor | Subscriber | Secondary actors | Library System |
| Description | Allows a subscriber to browse, select, and reserve available books. | | |
| Trigger | The authenticated subscriber accesses the Library System and selects one or more books to reserve. | | |
| Preconditions | * PRE-1. The subscriber must be authenticated. * PRE-2. The system must have an updated list of available books. | | |
| Postconditions | * POST-1. The selected books are reserved under the subscriber’s account. * POST-2. The availability status of the reserved books is updated in the system. | | |
| Normal flow | 1. Borrow books   1. The subscriber accesses the book catalog.  2. The subscriber searches for books using filters (searching on title and author).  3. The system displays a list of available books based on the search criteria.  4. The subscriber selects the books they wish to reserve and the quantity.  5. The system verifies the reservation conditions (availability, subscriber’s limit, etc.).  6. If all conditions are met, the system marks the books as reserved for the subscriber.  7. The system confirms the reservation and provides details on pickup instructions. | | |
| Alternative flows | 2.1 The subscriber doesn’t borrow any book  1. The subscriber accesses the book catalog.  2. The subscriber searches for books using filters or keywords.  3. The system displays a list of available books based on the search criteria.  4. The subscriber exit the Library System without borrowing any book. | | |
| Exceptions | **1.0.E1** Theselected book is no longer available If a selected book is no longer available, the system notifies the subscriber and suggests similar books. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC3 - Return Borrowed Books | | |
| Primary actor | Librarian | Secondary actors | Library System |
| Description | Allows the librarian to update book copies in the system to maintain an accurate inventory. | | |
| Trigger | The librarian initiates an inventory update. | | |
| Preconditions | * PRE-1. The system must have an interface for inventory management. * PRE-2. The librarian must be authenticated in the system and have the administrator rights. | | |
| Postconditions | * POST-1. The inventory is updated with the latest book availability. * POST-2. The changes are reflected in the book catalog for subscribers. * POST-3. The subscriber's borrowing record is updated. | | |
| Normal flow | 1.0 Update book copies.   1. The librarian logs into the library system. 2. The librarian selects the inventory management section. 3. The librarian chooses to update book copies. 4. The librarian selects a book and modifies number of copies. 5. The system applies the updates. 6. The system confirms the changes and updates the book catalog. | | |
| Alternative flows | 1.1 The librarian doesn’t update anything  The librarian exit the Library System without updating. | | |
| Exceptions | **1.0.**E1 If the subscriber tries to return a book not registered under their account, the system notifies the librarian. | | |

Descriptions of template fields:

* **ID and name:** Title should be descriptive and should usually begin with a verb, e.g. order, calculate, input, etc. ID can have any format but must be unique among all use cases.
* **Primary actor:** Person that wishes to accomplish a goal through the use of the system. Only a single primary actor per use case.
* **Secondary actors:** Actors that have an interest in the completion of the goal but that do not directly interact with the system.
* **Description:** Concise description of the purpose of the use case.
* **Trigger:** Condition internal or external to the system that prompts the use case to start.
* **Preconditions:** Conditions that must be true before the use case starts. Each should be labeled with an ID unique to the use case.
* **Postconditions:** Conditions that must be true after the use case ends normally. Each should be labeled with an ID unique to the use case.
* **Normal flow:** Detailed step-by-step description of the logical flow of the use case. It should describe an explicit two way interaction, with the system prompting for input and the actor responding accordingly. Each step should be numbered.
* **Alternative flows:** Flows that achieve the same goal as the normal flow but are expected to be less common or lower priority.
* **Exceptions:** Conditions that result in the normal flow ending prematurely due to an unrecoverable condition in the system. The condition that causes the flow should be clearly stated, as should be any other decisions that the actor must make in this situation.

## Examples

For a hypothetical *Cafeteria Ordering System*[[1]](#footnote-1):

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-1: Order a Meal | | |
| **Primary actor** | Patron | **Secondary actors** | Cafeteria Inventory System |
| **Description** | A Patron accesses the Cafeteria Ordering System from either the corporate intranet or external Internet, views the menu for a specific date, selects food items, and places an order for a meal to be picked up in the cafeteria or delivered to a specified location within a specified 15-minute time window. | | |
| **Trigger** | A Patron indicates that he wants to order a meal. | | |
| **Preconditions** | PRE-1. Patron is logged into COS.  PRE-2. Patron is registered for meal payments by payroll deduction. | | |
| **Postconditions** | POST-1. Meal order is stored in COS with a status of “Accepted.”  POST-2. Inventory of available food items is updated to reflect items in this order.  POST-3. Remaining delivery capacity for the requested time window is updated. | | |
| **Normal flow** | **1.0 Order a Single Meal**   1. Patron asks to view menu for a specific date. (see 1.0.E1, 1.0.E2) 2. COS displays menu of available food items and the daily special. 3. Patron selects one or more food items from menu. (see 1.1) 4. Patron indicates that meal order is complete. (see 1.2) 5. COS displays ordered menu items, individual prices, and total price, including taxes and delivery charge. 6. Patron either confirms meal order (continue normal flow) or requests to modify meal order (return to step 2). 7. COS displays available delivery times for the delivery date. 8. Patron selects a delivery time and specifies the delivery location. 9. Patron specifies payment method. 10. COS confirms acceptance of the order. 11. COS sends Patron an email message confirming order details, price, and delivery instructions. 12. COS stores order, sends food item information to Cafeteria Inventory System, and updates available delivery times. | | |
| **Alternative flows** | **1.1 Order multiple identical meals**   1. Patron requests a specified number of identical meals. (see 1.1.E1) 2. Return to step 4 of normal flow.   **1.2 Order multiple meals**   1. Patron asks to order another meal. 2. Return to step 1 of normal flow. | | |
| **Exceptions** | **1.0.E1 Requested date is today and current time is after today’s order cutoff time**  1. COS informs Patron that it’s too late to place an order for today.  2a. If Patron cancels the meal ordering process, then COS terminates use case.  2b. Else if Patron requests another date, then COS restarts use case.  **1.0.E2 No delivery times left**  1. COS informs Patron that no delivery times are available for the meal date.  2a. If Patron cancels the meal ordering process, then COS terminates use case.  2b. Else if Patron requests to pick the order up at the cafeteria, then continue with normal flow, but skip steps 7 and 8.  **1.1.E1 Insufficient inventory to fulfill multiple meal order**  1. COS informs Patron of the maximum number of identical meals he can order, based on current available inventory.  2a. If Patron modifies number of meals ordered, then return to step 4 of normal flow.  2b. Else if Patron cancels the meal ordering process, then COS terminates use case. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-5 Register for Payroll Deduction | | |
| **Primary actor** | Patron | **Secondary actors** | Payroll System |
| **Description** | Cafeteria patrons who use the COS and have meals delivered must be registered for payroll deduction. For noncash purchases made through the COS, the cafeteria will issue a payment request to the Payroll System, which will deduct the meal costs from the next scheduled employee payday direct deposit. | | |
| **Trigger** | Patron requests to register for payroll deduction, or Patron says yes when COS asks if he wants to register. | | |
| **Preconditions** | PRE-1. Patron is logged into COS. | | |
| **Postconditions** | POST-1. Patron is registered for payroll deduction. | | |
| **Normal flow** | **5.0 Register for Payroll Deduction**   1. COS asks Payroll System if Patron is eligible to register for payroll deduction. 2. Payroll System confirms that Patron is eligible to register for payroll deduction. 3. COS asks Patron to confirm his desire to register for payroll deduction. 4. If so, COS asks Payroll System to establish payroll deduction for Patron. 5. Payroll System confirms that payroll deduction is established. 6. COS informs Patron that payroll deduction is established. | | |
| **Alternative flows** | None | | |
| **Exceptions** | 5.0.E1 Patron is not a full time employee.  5.0.E2 Patron is already enrolled for payroll deduction. | | |

## Extra step: Traceability

For this extra step, you will add traceability information for each use case by adding a new field to the template:

|  |  |
| --- | --- |
| Method-level traces | <fully.qualified.ClassName>#<methodName>  ... |

Any method that implements the functionality described in the normal flow, alternative flow or exceptions should be included in this field. This means that the method that is initially executed and any methods of any classes that the work is delegated to should be included.

Examples for previous use cases:

UC-1:

|  |  |
| --- | --- |
| Method-level traces | my.company.ordering.MenuWidget#dateClicked  my.company.ordering.MenuWidget#completeOrder  my.company.ordering.InventoryInterface#checkInventory  ... |

UC-5:

|  |  |
| --- | --- |
| Method-level traces | my.company.payroll.PayrollInterface#checkEligibility  my.company.payroll.RegistrationForm#confirm  ... |

1. Examples adapted from Wiegers, K. E. & Beatty, J. (2013) Software requirements . 3rd ed. Redmond, WA: Microsoft Press. [↑](#footnote-ref-1)