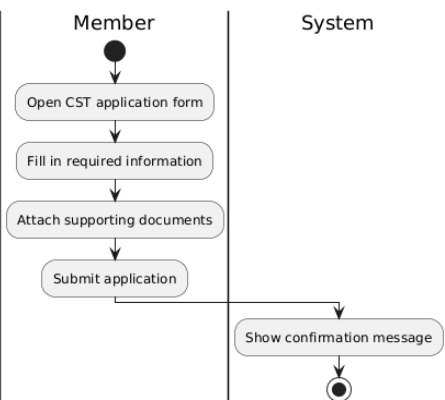
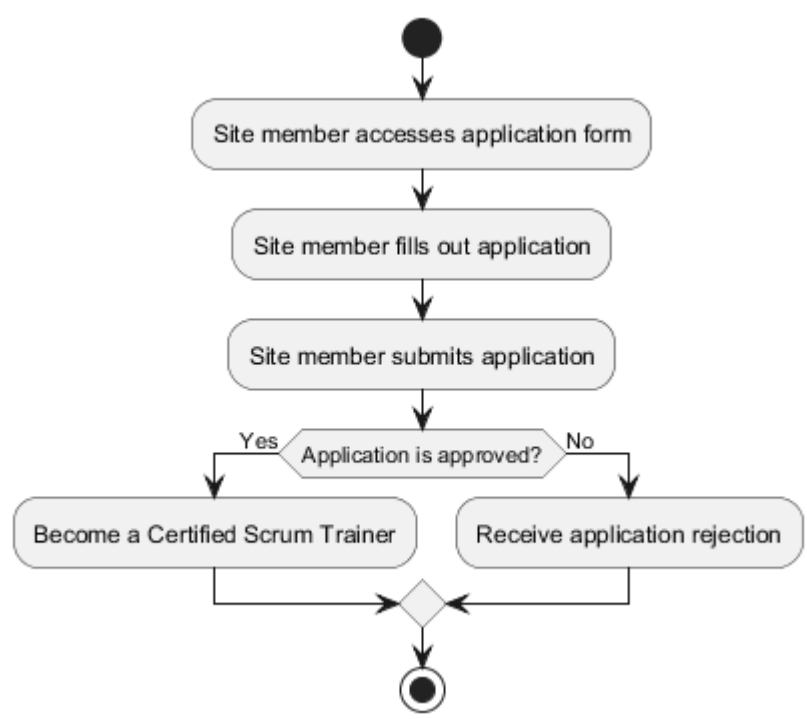
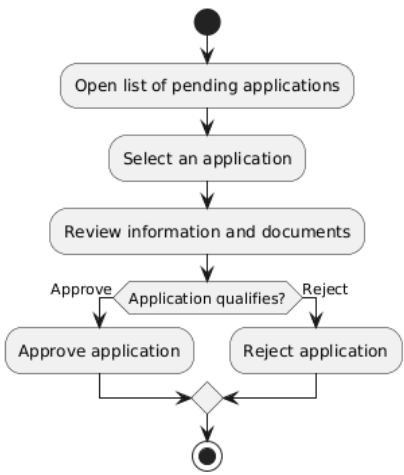
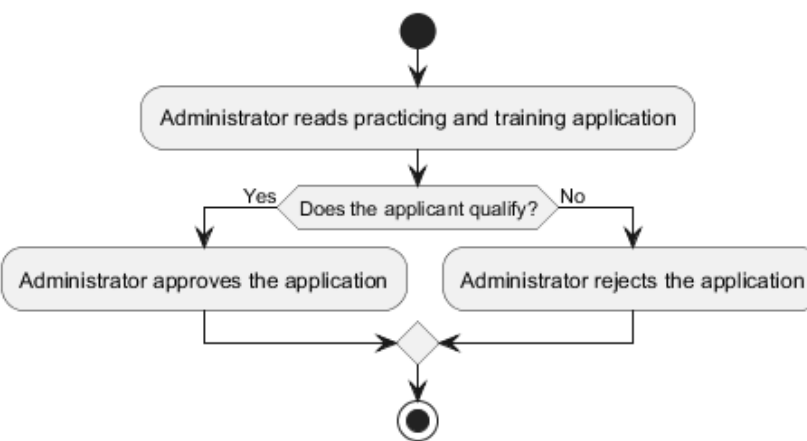

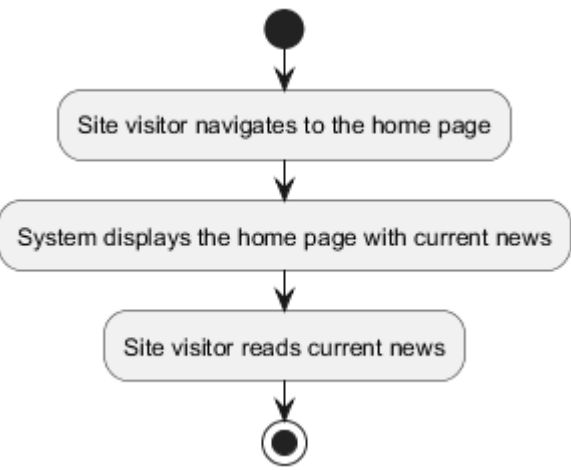
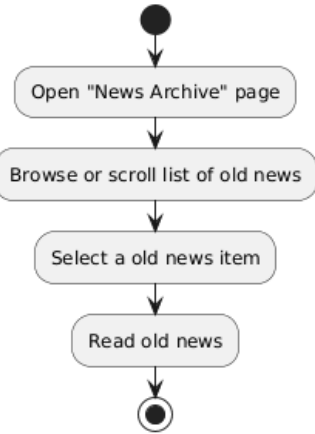
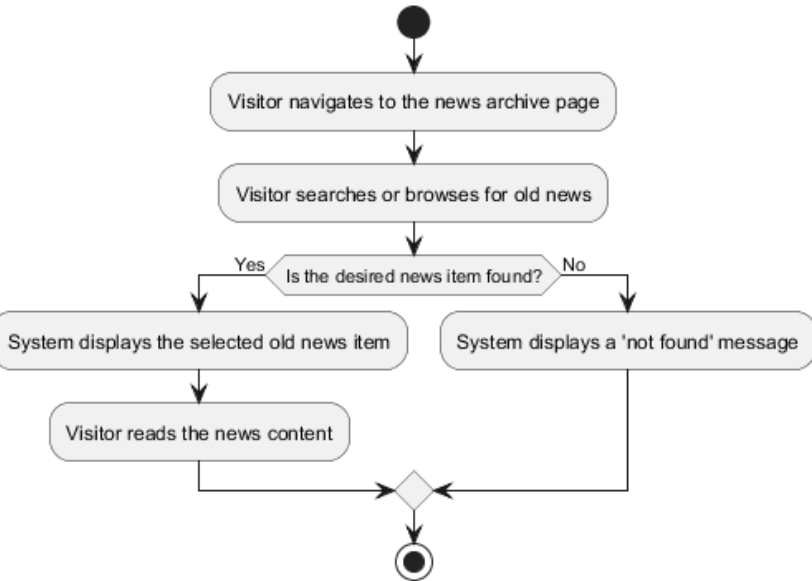
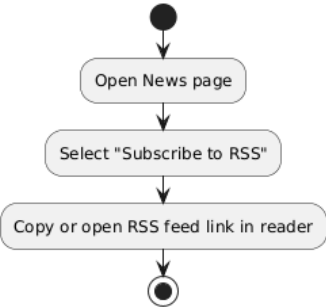
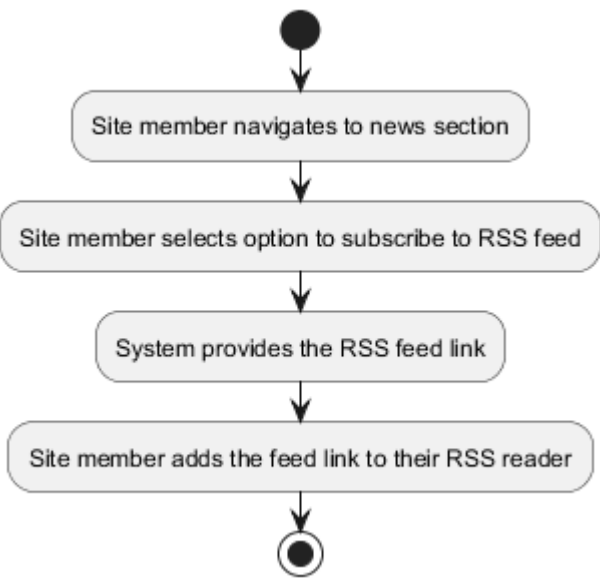
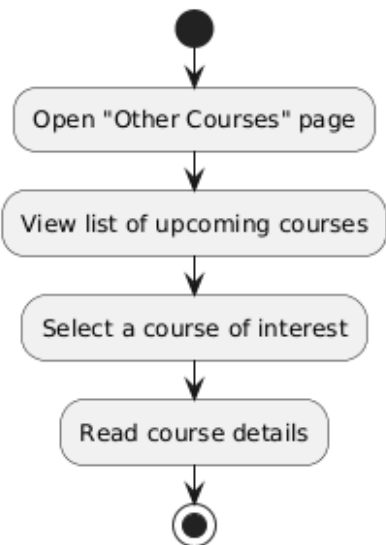
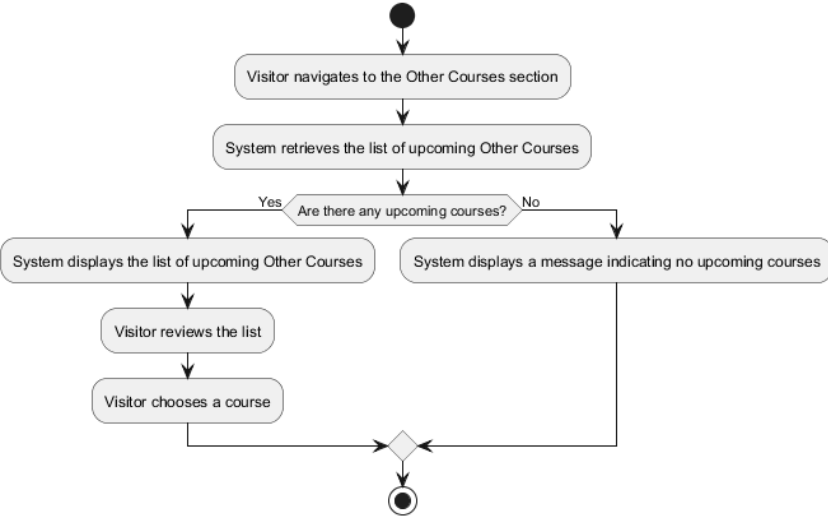


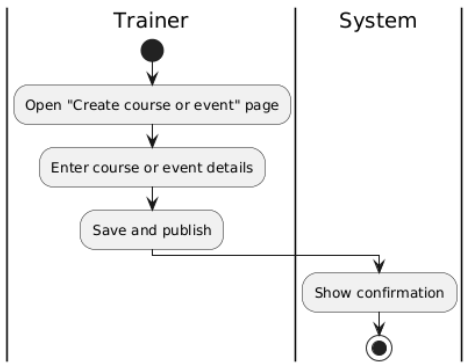
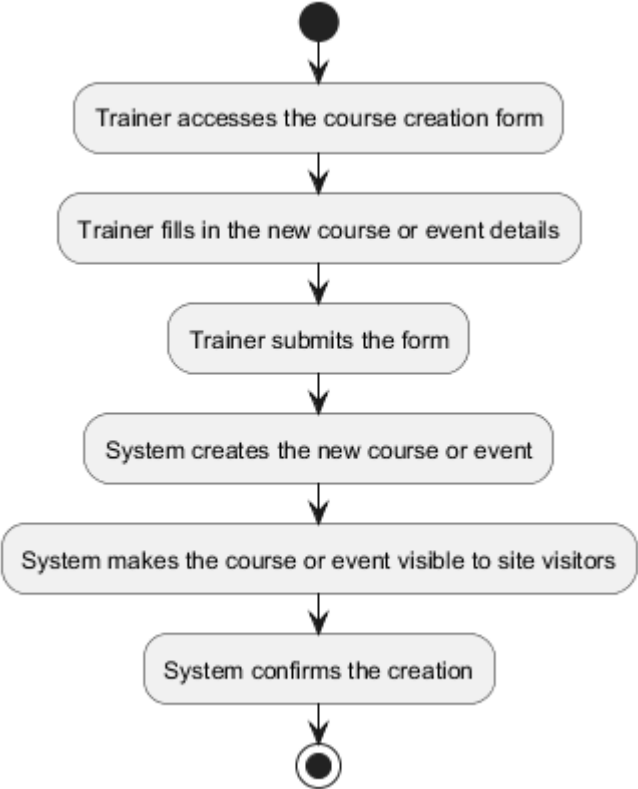
DIAGRAMAS DE REFERENCIA Y DIAGRAMAS GENERADOS

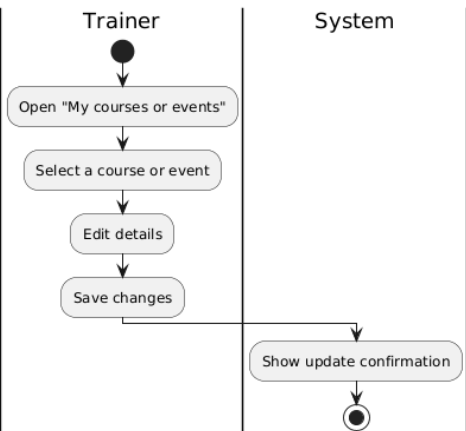
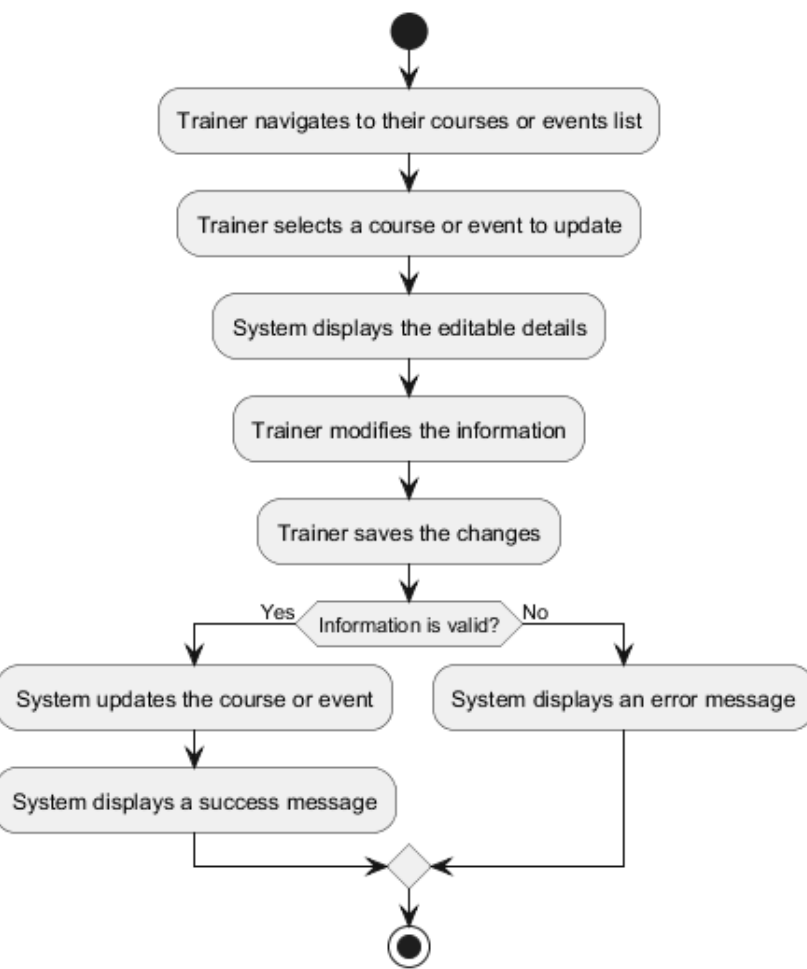
ID	Diagramas de Referencia (Humano)		Diagramas Generados (Herramienta)	
	Código en PlantUML	Diagrama en PlantUML	Código en PlantUML	Diagrama en PlantUML
HU-AC01	<pre>@startuml title Apply for Certified Scrum Practitioner (CSP) [Member] start :Open "CSP Application"; [System] :Display application form; [Member] :Fill required fields; :Attach evidence (files); :Preview application; :Submit; [System] if (Input valid & files ok?) then (Yes) :Persist application; :Assign trackingId; :Queue confirmation email; :Show success message; stop else (No) :Show inline errors; :Keep entered data; endif @enduml</pre>	<pre>sequenceDiagram participant Member participant System Member->>System: Open "CSP Application" System->>Member: Display application form Member->>System: Fill required fields Member->>System: Attach evidence (files) Member->>System: Preview application Member->>System: Submit System->>System: Input valid & files ok? System-->>System: Yes System->>System: Persist application System->>System: Assign trackingId System->>System: Queue confirmation email System->>System: Show success message System-->>System: No System->>System: Show inline errors System->>System: Keep entered data</pre>	<pre>@startuml start :Access the application form for Certified Scrum Practitioner; :Fill out the application; :Submit the application; stop @enduml</pre>	<pre>graph TD Start(()) --> A[Access the application form for Certified Scrum Practitioner] A --> B[Fill out the application] B --> C[Submit the application] C --> End((()))</pre>

HU-AC02	<p>@startuml [Member] start :Open CST application form; :Fill in required information; :Attach supporting documents; :Submit application; [System] :Show confirmation message; stop @enduml</p>	 <pre>graph TD subgraph Member Start(()) --> U1[Open CST application form] U1 --> U2[Fill in required information] U2 --> U3[Attach supporting documents] U3 --> U4[Submit application] end subgraph System U4 --> U5[Show confirmation message] U5 --> End(()) end</pre>	<p>@startuml start :Site member accesses application form; :Site member fills out application; :Site member submits application; if (Application is approved?) then (Yes) :Become a Certified Scrum Trainer; else (No) :Receive application rejection; endif stop @enduml</p>	 <pre>graph TD Start(()) --> U1[Site member accesses application form] U1 --> U2[Site member fills out application] U2 --> U3[Site member submits application] U3 --> D1{Application is approved?} D1 -- Yes --> U4[Become a Certified Scrum Trainer] D1 -- No --> U5[Receive application rejection] U4 --> D2{ } U5 --> D2 D2 --> End(())</pre>
HU-AC03	<p>@startuml start :Open list of pending applications; :Select an application; :Review information and documents; if (Application qualifies?) then (Approve) :Approve application; else (Reject) :Reject application; endif stop @enduml</p>	 <pre>graph TD Start(()) --> U1[Open list of pending applications] U1 --> U2[Select an application] U2 --> U3[Review information and documents] U3 --> D1{Application qualifies?} D1 -- Approve --> U4[Approve application] D1 -- Reject --> U5[Reject application] U4 --> D2{ } U5 --> D2 D2 --> End(())</pre>	<p>@startuml start :Administrator reads practicing and training application; if (Does the applicant qualify?) then (Yes) :Administrator approves the application; else (No) :Administrator rejects the application; endif stop @enduml</p>	 <pre>graph TD Start(()) --> U1[Administrator reads practicing and training application] U1 --> D1{Does the applicant qualify?} D1 -- Yes --> U2[Administrator approves the application] D1 -- No --> U3[Administrator rejects the application] U2 --> D2{ } U3 --> D2 D2 --> End(())</pre>

HU-AC04	<p>@startuml start :Open homepage; :View current news headlines; :Select a news item; :Read news; stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1([Open homepage]); UC1 --> UC2([View current news headlines]); UC2 --> UC3([Select a news item]); UC3 --> UC4([Read news]); UC4 --> End((()))</pre>	<p>@startuml start :Site visitor navigates to the home page; :System displays the home page with current news; :Site visitor reads current news; stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1([Site visitor navigates to the home page]); UC1 --> UC2([System displays the home page with current news]); UC2 --> UC3([Site visitor reads current news]); UC3 --> End((()))</pre>
HU-AC05	<p>@startuml start :Open "News Archive" page; :Browse or scroll list of old news; :Select a old news item; :Read old news; stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1([Open "News Archive" page]); UC1 --> UC2([Browse or scroll list of old news]); UC2 --> UC3([Select a old news item]); UC3 --> UC4([Read old news]); UC4 --> End((()))</pre>	<p>@startuml start :Visitor navigates to the news archive page; :Visitor searches or browses for old news; if (Is the desired news item found?) then (Yes) :System displays the selected old news item; :Visitor reads the news content; else (No) :System displays a 'not found' message; endif stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1([Visitor navigates to the news archive page]); UC1 --> UC2([Visitor searches or browses for old news]); UC2 --> Decision{Is the desired news item found?}; Decision -- Yes --> UC3([System displays the selected old news item]); Decision -- No --> UC4([System displays a 'not found' message']); UC3 --> UC5([Visitor reads the news content]); UC4 --> Merge(()); UC5 --> Merge; Merge --> End((()))</pre>

HU-AC06	<p>@startuml start :Open News page; :Select "Subscribe to RSS"; :Copy or open RSS feed link in reader; stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1[Open News page]; UC1 --> UC2[Select "Subscribe to RSS"]; UC2 --> UC3[Copy or open RSS feed link in reader]; UC3 --> End((()))</pre>	<p>@startuml start :Site member navigates to news section; :Site member selects option to subscribe to RSS feed; :System provides the RSS feed link; :Site member adds the feed link to their RSS reader; stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1[Site member navigates to news section]; UC1 --> UC2[Site member selects option to subscribe to RSS feed]; UC2 --> UC3[System provides the RSS feed link]; UC3 --> UC4[Site member adds the feed link to their RSS reader]; UC4 --> End((()))</pre>
HU-AC07	<p>@startuml start :Open "Other Courses" page; :View list of upcoming courses; :Select a course of interest; :Read course details; stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1[Open "Other Courses" page]; UC1 --> UC2[View list of upcoming courses]; UC2 --> UC3[Select a course of interest]; UC3 --> UC4[Read course details]; UC4 --> End((()))</pre>	<p>@startuml start :Visitor navigates to the Other Courses section; :System retrieves the list of upcoming Other Courses; if (Are there any upcoming courses?) then (Yes) :System displays the list of upcoming Other Courses; :Visitor reviews the list; :Visitor chooses a course; else (No) :System displays a message indicating no upcoming courses; endif stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1[Visitor navigates to the Other Courses section]; UC1 --> UC2[System retrieves the list of upcoming Other Courses]; UC2 --> Decision{Are there any upcoming courses?}; Decision -- Yes --> UC3[System displays the list of upcoming Other Courses]; UC3 --> UC4[Visitor reviews the list]; UC4 --> UC5[Visitor chooses a course]; UC5 --> End(()); Decision -- No --> UC6[System displays a message indicating no upcoming courses]; UC6 --> End</pre>

HU-AC08	<p>@startuml [Trainer] start :Open "Create course or event" page; :Enter course or event details; :Save and publish; [System] :Show confirmation; stop @enduml</p>	 <pre>graph TD subgraph Trainer Start(()) --> Open[Open "Create course or event" page] Open --> Enter[Enter course or event details] Enter --> Save[Save and publish] end subgraph System Save --> Show[Show confirmation] Show --> End((())) end</pre>	<p>@startuml start :Trainer accesses the course creation form; :Trainer fills in the new course or event details; :Trainer submits the form; :System creates the new course or event; :System makes the course or event visible to site visitors; :System confirms the creation; stop @enduml</p>	 <pre>graph TD Start(()) --> A[Trainer accesses the course creation form] A --> B[Trainer fills in the new course or event details] B --> C[Trainer submits the form] C --> D[System creates the new course or event] D --> E[System makes the course or event visible to site visitors] E --> F[System confirms the creation] F --> End((()))</pre>
---------	---	--	---	--

<p>HU-AC09</p>	<p>@startuml Trainer start :Open "My courses or events"; :Select a course or event; :Edit details; :Save changes; [System] :Show update confirmation; stop @enduml</p>	 <pre> sequenceDiagram participant Trainer participant System Trainer->>Trainer: Open "My courses or events" Trainer->>Trainer: Select a course or event Trainer->>Trainer: Edit details Trainer->>Trainer: Save changes Trainer->>System: System->>System: Show update confirmation </pre>	<p>@startuml start :Trainer navigates to their courses or events list; :Trainer selects a course or event to update; :System displays the editable details; :Trainer modifies the information; :Trainer saves the changes; if (Information is valid?) then (Yes) :System updates the course or event; :System displays a success message; else (No) :System displays an error message; endif stop @enduml</p>	 <pre> graph TD Start(()) --> A[Trainer navigates to their courses or events list] A --> B[Trainer selects a course or event to update] B --> C[System displays the editable details] C --> D[Trainer modifies the information] D --> E[Trainer saves the changes] E --> F{Information is valid?} F -- Yes --> G[System updates the course or event] F -- No --> H[System displays an error message] G --> I[System displays a success message] I --> J{ } H --> J J --> End((())) </pre>
----------------	--	--	---	---

