April 4, 2025

Yubiq Studio Training & Developer Manual

Rev. 5.1

|  |  |  |
| --- | --- | --- |
| WRITTEN BY: |  |  |
| VERIFIED BY: |  |  |
| APPROVED BY: |  |  |

# Document Change Log

|  |  |  |
| --- | --- | --- |
| Rev | Description | Dates |
| 5.0.14 | Work in Progress | TBD |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Index

[Document Change Log 3](#_Toc198300034)

[Index 4](#_Toc198300035)

[Introduction 6](#_Toc198300036)

[The ST.A.R. Methodology. 7](#_Toc198300037)

[Stages (Phases) 7](#_Toc198300038)

[Actions (Actions) 7](#_Toc198300039)

[Roles 7](#_Toc198300040)

[WorkItem (Practice) 7](#_Toc198300041)

[STAR representation example 7](#_Toc198300042)

[Yubiq Studio (YS) 10](#_Toc198300043)

[Installation 11](#_Toc198300044)

[The SandBox 12](#_Toc198300045)

[Training Project: Flight 13](#_Toc198300046)

[Process description 13](#_Toc198300047)

[The necessary steps 14](#_Toc198300048)

[Project and Workflow 15](#_Toc198300049)

[Modeling a Project 15](#_Toc198300050)

[Modeling a Workflow 16](#_Toc198300051)

[Roles 17](#_Toc198300052)

[Static Roles 17](#_Toc198300053)

[The Dynamic Roles 17](#_Toc198300054)

[Modeling of Static and Dynamic Roles 18](#_Toc198300055)

[SandBox Data: Users 19](#_Toc198300056)

[Business Object 20](#_Toc198300057)

[Business Object Examples 20](#_Toc198300058)

[Business Object Modeling 21](#_Toc198300059)

[C# coding of Business Objects 22](#_Toc198300060)

[Business Object (EnumValue) 24](#_Toc198300061)

[Business Object Modeling (EnumValue) 24](#_Toc198300062)

[SandBox Data: Business Objects 25](#_Toc198300063)

[SandBoxData/Business Object Modeling. 26](#_Toc198300064)

[Workitem 27](#_Toc198300065)

[WorkItem modeling 27](#_Toc198300066)

[WorkItem c# coding 28](#_Toc198300067)

[Process Map 29](#_Toc198300068)

[WorkItem modeling 29](#_Toc198300069)

[Forms 32](#_Toc198300070)

# Introduction

This document aims to supplement the Yubiq BPM Solution Development Manual with the related training course

After an overview of the STAR methodology, which is the basis of BPM modeling, we will take an example business flow, such as **a request for airline tickets**, and, implementing it step-by-step, go on to detail all aspects of its development.

Yubiq Studio (from now YS) is a development tool aimed at two categories of specialists:

**Business analysts**.

YS, with its "Modeller" mode, allows the definition of basic structures without writing code (No-Code) such as: **BusinessObjects, WorkItem, Stages, Actions,**

**C# developers.**

The "Developer" mode then allows you to complete the remaining structures such as **CodeLibs** that go to contain machine code for data processing and integrations or **CustomViews** to create custom gr afici objects.

Before delving into development with YS, let us introduce the basic concepts of the STAR methodology. This is not a notation or formalism like BPMN, but rather a decomposition of a business process into STages, Actions and Roles.

# The ST.A.R. Methodology.

The STAR (Stages, Action, and Roles) methodology was developed in Palo Alto, California, and is a Business Process Management (BPM) approach aimed at improving the efficiency and effectiveness of business processes.

It represents a structured approach to implementing Business Process Management (BPM) solutions that emphasizes the importance of clarity in the steps, actions to be taken, and roles of participants. Here is an overview of the main elements of the methodology:

## Stages (Phases)

Stages represent the different stages of the business process management process. Good examples may be:

* **New Practices**
* **Practices to be worked on**
* **Practices in error**
* **Archive**

## Actions (Actions)

Actions refer to the specific activities to be performed at each stage. Good examples may be:

* **Edit data**
* **Advance to the Next Stage**
* **Assign to other user**

## Roles

Clearly defining roles is critical to the success of the methodology. Each participant has specific responsibilities, and their involvement is essential for successful implementation. Roles may include:

* **Applicant**
* **Approver**
* **Travel Office**
* **Director**

## WorkItem (Practice)

In the STAR methodology, there is also the concept of a **WorkItem**, which represents a specific unit of work assigned to an individual or group within a process. A WorkItem can represent any practice, activity or task that must be completed in order for the process to continue or achieve its objectives.

## STAR representation example

If we take a BPM project for "Requesting airline tickets" as an example, an example of **Stages** and **Actions** might be as follows:

1. **Beginning**:
   * Creation of the flight request by the applicant.
   * Collection of necessary information (applicant's data, flight details, etc.).
2. **Preliminary Verification**:
   * Checking the information entered.
   * Checking the suitability of the request (e.g., budget, company policies, availability).
3. **Approval**:
   * Forwarding the request to the designated approver (supervisor or manager).
   * The request is in pending approval status.
   * The decision maker can approve, reject, or request additional information.
4. **Flight Search**:
   * Contact providers (travel agencies or reservation systems) to research flight options.
   * Collection of available flight options and their costs.
5. **Presentation Options**:
   * Presentation of flight options to the applicant.
   * Collection of feedback or final choice from the applicant.
6. **Reservation**:
   * Confirmation and booking of the selected flight.
   * Issuing tickets and communicating the information to the applicant.
7. **Preparation for the Journey**:
   * Sending trip information (itinerary, registration, documentation).
   * Verification of compliance with travel regulations (e.g., documentation, luggage, etc.).
8. **Execution of the Journey**:
   * The applicant embarks on the journey.
   * Monitoring the progress of the request (any problems or changes).
9. **Closure**:
   * Return of the traveler.
   * Evaluation of the travel experience (applicant feedback).
   * Closing of application and recording of expenses (reimbursement, if applicable).
10. **Post-travel analysis**:
    * Collect data on actual expenditures versus budget.
    * Analysis of workflow performance for future optimizations.

In a request-flight practice, a **WorkItem** can be defined with several attributes to ensure effective handling of the request. Here is a list of the most common attributes that might be associated with a WorkItem in this context:

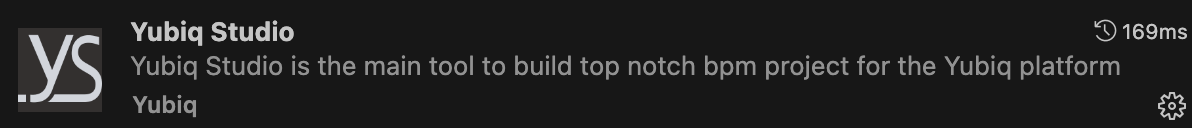
1. **WorkItem ID**: A unique identifier for each flight request.
2. **Creation** Date: The date and time when the flight request was generated.
3. **Requestor**: Name and contact details of the person making the request.
4. **Type of Flights**: Specifies whether it is a domestic flight, international, round trip, etc.
5. **Destination**: Place to which you wish to fly.
6. **Departure** Date and Time: The desired departure date and time.
7. **Return Date and Time** (if applicable): The desired return date and time.
8. **Number of Passengers**: Amount of people who will travel.
9. **Class of Flight**: Preferred class of travel (economy, business, first class).
10. **Reason for Travel**: Description of the reason for which the request is made (e.g., business, vacation, urgent).
11. **Request Status**: The current status of the WorkItem (e.g., "Pending Approval," "Approved," "Rejected," "Completed").
12. **Due Date**: A deadline by which the request must be processed.
13. **Additional Notes**: Space for comments or relevant information submitted by the applicant.
14. **Assignee**: Person or team responsible for handling the flight request.
15. **Change History**: Record of changes made to the request (status changes, updates, etc.).

In conclusion, the STAR methodology represents an innovative and effective approach to business process management. Through the clear definition of steps, actions and roles, it is possible to optimize workflow, improve operational efficiency and ensure greater transparency and accountability within the organization. Its application in BPM projects such as booking meeting rooms or requesting airline tickets demonstrates the versatility and usefulness of this method. With Yubiq Studio and the STAR methodology, companies can simplify and automate their processes, facilitating the management of daily activities and promoting a culture of continuous improvement. The integration of these tools and methodologies provides a significant competitive advantage, contributing to the success and sustainable growth of the organization.

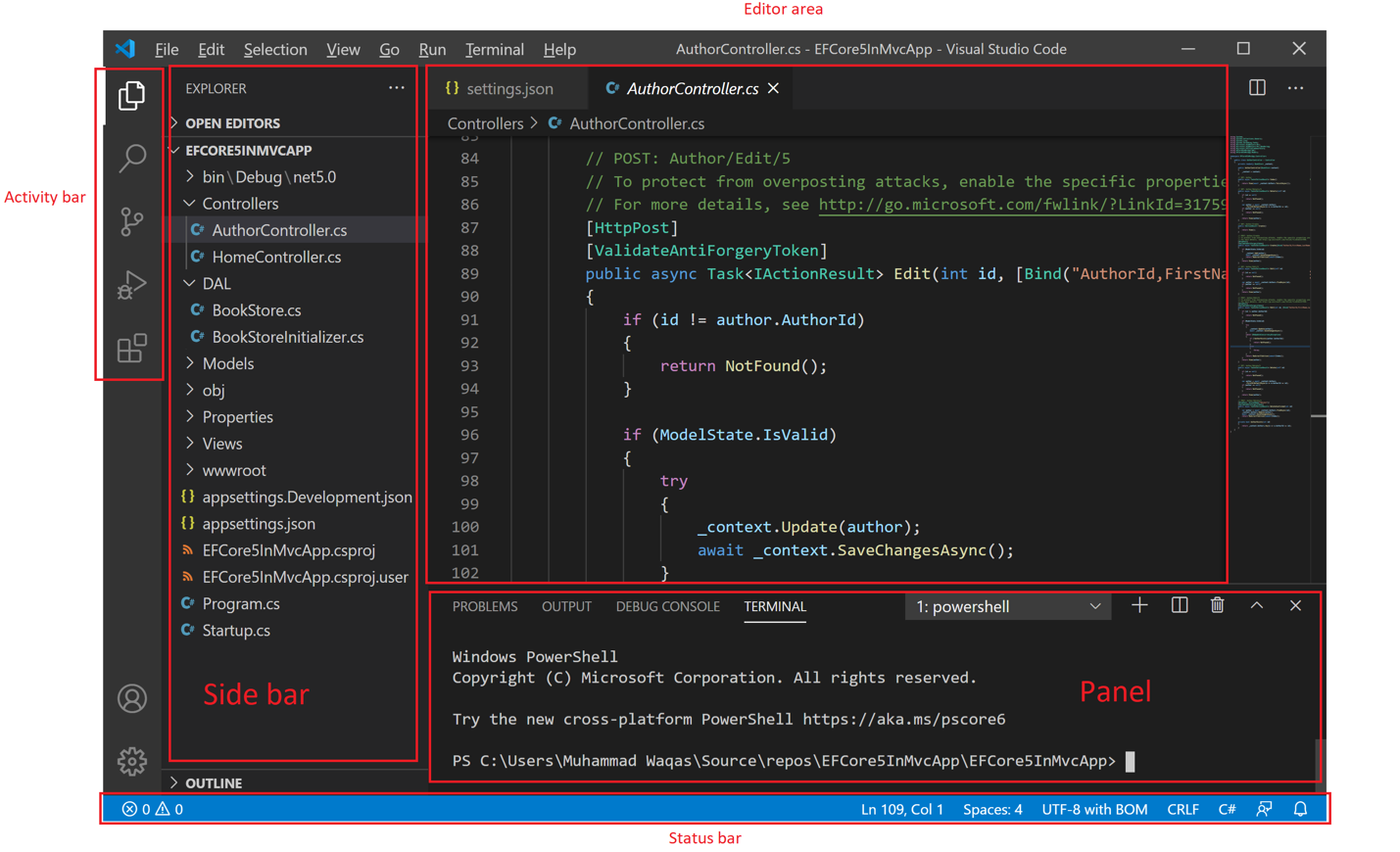
## Yubiq Studio (YS)

Yubiq Studio is based entirely on **Visual Studio Code**, (VSC or vscode), is a free source code editor developed by Microsoft for **Windows, macOS and Linux.** Features include debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and built-in version control with Git. With nearly **75 percent market share and 14 million users**, VSCode is the most popular IDE available (source Wikipedia).

Because VSC is a generic IDE there are over 30,000 extensions in its store that allow you to customize virtually anything in the IDE. YS is formally a Visual Studio Code extension, which can be downloaded and installed from the Microsoft store.



Yubiq Studio is a versatile and advanced development tool designed to streamline business processes and improve operational efficiency. Its user-friendly and intuitive interface allows developers to create, manage and automate WorkFlows without having to resort to external solutions or scripts. With its perfectly structured and interrelated Stages, Actions, Roles and WorkItems, Yubiq Studio provides an integrated and consistent development experience.



During training with Yubiq Studio, users can explore and use a unique "functional tree" that facilitates navigation and interaction with Project objects. WorkFlow Forms are fully implementable with YS, using both standard graphical objects, called CoreViews, and the CustomViews mechanism to create custom graphical objects. This makes Yubiq Studio a powerful and flexible tool, suited to meet the specific needs of each organization.

In addition, Yubiq Studio supports integration with extensions such as Microsoft .NET C# and GitHub Copilot, which add additional development and AI assistance capabilities, making the development process even more efficient and productive.

With Yubiq Studio, companies can simplify and automate their processes, facilitating the management of daily activities and promoting a culture of continuous improvement.

## Installation

Once you have installed VSC on a Windows or MAC environment (Linux is not supported at the moment) proceed with the installation of the .NET 8 SDK: <https://dotnet.microsoft.com/en-us/download/dotnet/8.0>

And thus of the following extensions of VSC.

**Microsoft .NET C#**: The basic extensions of .NET C# development.

Immagine che contiene testo, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.

**GitHub Copilot**: For adding AI during development. Strongly recommended especially if you have a full user license!

Immagine che contiene testo, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

**Utility extensions:** We also recommend installing the following extensions that provide useful functionality during development:

**Immagine che contiene testo, schermata, Carattere, software

Il contenuto generato dall'IA potrebbe non essere corretto.**

**Yubiq Studio: Finally we install** the extension for Yubiq Studio!

Immagine che contiene testo, schermata, Carattere, nero

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene testo, schermata, Carattere, design

Il contenuto generato dall'IA potrebbe non essere corretto.At this point VSC is ready for the creation of a new project

During training with Yubiq Studio, in addition to **Stages**, **Actions**, **Roles** and the **WorkItem** you will discover other objects that are perfectly structured, related to each other and explorable from a single "functional tree."

All **Project** development is done exclusively with YS without any use of external Solution or scripts.

All the **Forms** in a **WorkFlow** are made with YS and all the standard graphic objects called **CoreViews** made available can be used. If you need to make a custom graphical object we are helped by the **CustomViews** mechanism within the standard Forms.

## The SandBox

The SandBox is an isolated development environment that allows a project to be simulated and tested without affecting the actual production system. This tool is essential for the initial phase of development, as it allows you to work on various aspects of the project, such as analysis, flow modeling, creation of ancillary data structures, and design of interaction forms. In addition, all code that does not depend on the Yubiq production environment can be written and tested in the SandBox.

Immagine che contiene schermata, software, testo, Icona del computer

Il contenuto generato dall'IA potrebbe non essere corretto.Using the SandBox, about 90 percent of the design can be done locally, thus providing greater efficiency and speed in the development phase. Prototype demonstration phases to the customer can also be carried out through the SandBox, which can be made "portable" to facilitate presentations in different contexts.

Another advantage of the SandBox environment is the ability to use simulated BusinessObjects to test the operation of the project. This allows you to verify proper integration and interaction between the various components of the system before moving on to the production phase.

The SandBox thus provides a safe and controlled environment for design optimization and verification.

YS allows:

* Launch the SandBox locally on your Browser at any time during solution development. Great for constant testing and local debugging
* Export a ZIP file with the SandBox to run on any Windows or Mac client.
* Publish the SandBox on the "SandBox Server" so that anyone in possession of the link can run it.

# Training Project: Flight

Project Flight coordinates the airline ticket application process, including the various approvals and issuance of tickets by the travel office.

We will use this example process to illustrate how to build a BPM solution with Yubiq. We recommend that you keep the complete Flight solution open in one window of VSC and your solution that you will create step by step following this guide in the other.

First, interested employees must submit a request with detailed information regarding the destination, dates, and reasons for travel. This request is then submitted to managers for approval, who assess the need for and compliance of the expenses with company policy. Once approval is obtained, the travel office proceeds with the selection of the most cost-effective flight options suitable for the travelers' time needs. Finally, after the employee has confirmed the choice, tickets are issued and delivered to the applicant, thus completing the process.

## Thumbnail Image Crea un'immagine che rappresenta un processo di business per la richiesta di biglietti aerei. Includi elementi come un richiedente che compila una richiesta di volo, un manager che approva la richiesta, un ufficio viaggi che prenota il biglietto e un viaggiatore che intraprende il viaggio. Mostra le fasi del processo come creazione della richiesta, verifica preliminare, approvazione, ricerca volo, presentazione opzioni, prenotazione, preparazione al viaggio, esecuzione del viaggio, chiusura e analisi post-viaggio. Utilizza uno stile professionale e chiaro, con icone rappresentative per ogni fase e ruoli definiti.Process description

1. Both Employee and Director users have the functionality to request an airline ticket and enter the necessary information.
2. If he is an Employee:
   1. your request is subject to the approval of your Manager
   2. The Manager may ask clarifying questions to the Employee who may possibly waive the flight
   3. Once approval is obtained, the file goes to the travel office
3. If it is a Director the file goes directly to the Travel Office.
4. Travel office books the ticket and passes the file to the applicant
5. The applicant, within three days of the departure date, can request a flight change directly to the Travel Office.
6. The applicant, once the ticket is obtained, can:
   1. give indication of having made the flight
   2. Giving up the trip
   3. If more than one week passes from the departure date, the ticket is automatically "waived"

Let's start by defining the basic objects for this project. That is, the **Roles**, The **Business Objects** and the **WorkItem** also we will have to create the simulated data for the **SandBox** for the **Users** and the different **BusinessObjects (BO)**.

### The necessary steps

In order to carry out the process described we will need to perform the steps shown in the table. For each step we indicate the most suitable professional figure between the **Modeller** and the **Developer**.

|  |  |  |
| --- | --- | --- |
| **Item** | **Modeller** | **Developer** |
| Project and Workflow | Badge Tick1 con riempimento a tinta unita |  |
| Roles | Badge Tick1 con riempimento a tinta unita |  |
| Business Objects | Badge Tick1 con riempimento a tinta unita |  |
| WorkItem | Badge Tick1 con riempimento a tinta unita  Basic attributes | Badge Tick1 con riempimento a tinta unita  Advanced attributes |
| SandBox Data | Badge Tick1 con riempimento a tinta unita | Badge Tick1 con riempimento a tinta unita |
| Process (Stages and Roles) | Badge Tick1 con riempimento a tinta unita  Definition | Badge Tick1 con riempimento a tinta unita  Code |
| Forms | Badge Tick1 con riempimento a tinta unita  Basic definition | Badge Tick1 con riempimento a tinta unita  Layouts, Lookups, Validation |
| CodeLibs | Badge Croce con riempimento a tinta unita | Badge Tick1 con riempimento a tinta unita |
| CustomView | Badge Croce con riempimento a tinta unita | Badge Tick1 con riempimento a tinta unita |

In the next chapters we will go into detail on each of the listed steps always making sure to describe first the practical example of the "Flight" project and then the implementation aspects.

## Thumbnail Image Crea un'immagine che rappresenti un workflow molto semplice. Il workflow dovrebbe includere fasi chiare e ben definite come 'Inizio', 'Verifica Preliminare', 'Approvazione', 'Ricerca Volo', 'Presentazione Opzioni', 'Prenotazione', 'Preparazione al Viaggio', 'Esecuzione del Viaggio', 'Chiusura', e 'Analisi Post-viaggio'. Utilizza icone e grafici per rappresentare ogni fase e azione, con frecce che mostrano il flusso da una fase all'altra. Usa colori chiari e uno stile minimalista per rendere l'immagine facile da comprendere. Includi anche i ruoli principali come 'Richiedente', 'Approvatore', 'Ufficio Viaggi', e 'Direttore' in modo che siano chiaramente identificabili.Project and Workflow

Creating a Project is the first step we need to take in order to create the initial structure of the solution. Once created we can go on to define one or more Workflows that will make up the solution.

We will then create the Project **Flight** first and then the Workflow **Flight**.

### ****Modeling a Project****

**Immagine che contiene Personaggio immaginario, Cartoni animati, illustrazione, Animazione

Il contenuto generato dall'IA potrebbe non essere corretto.**

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Example** |
| **1** | **Select the YS extension on the Activity Bar and then click on "New Project"**  **You will be asked to indicate a parent folder where you can create the new Project folder**  **Then enter the name of the new Project, which we will call Flight**  **Contextually, a folder with the same name will be created under the selected folder** |  |
| **2** | **Locate in the Side Bar the YUBIQ STUDIO tree, which should show up as in the figure.**  **One of the first branches is precisely the Workflows**  **Note that it has no Workflow defined** |  |

### ****Modeling a Workflow****

**Immagine che contiene Personaggio immaginario, Cartoni animati, illustrazione, Animazione

Il contenuto generato dall'IA potrebbe non essere corretto.**

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Example** |
| **1** | **Locate the Workflows branch and click on the icon**  **Enter Workflow name without spaces Flight** |  |
| **2** | **YS will generate the initial Workflow structure with:**   * **The Action of Creation** * **A First Stage** * **The final stage of Archive** * **An empty Minimal Form**   **This aspect allows for the almost immediate execution of the SandBox** |  |
| **4** | **Simultaneously with the generation of the necessary code, the process design is also generated.**  **Clicking on the icon in the newly created workflow branch shows this** |  |

We will continue with the process definition after creating the main basic objects.

## Thumbnail Image Crea un'immagine che rappresenta i diversi ruoli utente nella metodologia STAR di Yubiq Studio. Includi i ruoli di Richiedente, Approvatore, Ufficio Viaggi e Direttore. Utilizza icone o figure stilizzate per ogni ruolo, con colori distinti per differenziarli. Mostra una struttura gerarchica o un flusso di lavoro che evidenzi come i ruoli interagiscono tra loro. Lo stile dell'immagine dovrebbe essere professionale e moderno, con un design pulito e chiaro. Utilizza font leggibili e un layout che faciliti la comprensione visiva dei ruoli e delle loro responsabilità.Roles

We describe the Roles that users are to have by distinguishing between static and dynamic ones. Recall that by static role we denote a fixed group of users while a user gets a dynamic role based on the rules of the process.

### Static Roles

Static roles are assigned to a fixed group of users within a system or solution. These roles do not change based on user actions or circumstances, but remain constant over time. For example, within an enterprise solution, we might have two main static roles:

* Employee: All employees of the company belong to this role. These users have access to certain resources and functionality defined by their employee position.
* Director: Company directors belong to this role. They have higher privileges and access than the Employee, in line with their management responsibilities.

In addition, corporate travel office users can belong to the static **TravelDept** role, which grants them specific access for corporate travel management.

### The Dynamic Roles

Dynamic roles, in contrast, are assigned to users based on certain conditions or actions within the process. These roles can change and are not fixed, depending on operational needs. For example:

* **Traveler**: When a user makes a new airline ticket request, he or she automatically acquires the dynamic role of Traveler. This role allows the user to access specific features related to travel management.
* **Manager**: When a user acquires the Traveler role, his or her direct manager is identified, who acquires the dynamic role of Manager. To implement this logic, the user definition is extended with the ManagedBy attribute to identify the manager.

**Immagine che contiene testo, schermata, Carattere, design

Il contenuto generato dall'IA potrebbe non essere corretto.**

### **Modeling of Static and Dynamic Roles**

**Immagine che contiene Personaggio immaginario, Cartoni animati, illustrazione, Animazione

Il contenuto generato dall'IA potrebbe non essere corretto.**

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Example** |
| **1** | **Position yourself on the YS tree and select the "Roles" branch. The Static and Dynamic branches will appear. Select the sub-branch of interest** | **Immagine che contiene testo, schermata, Carattere, Elementi grafici  Il contenuto generato dall'IA potrebbe non essere corretto.** |
| **2** | **Select the sub-branch of interest. For example,**  **You then click on the icon and in the top bar you are asked for the name to be given to it**  **The procedure for Dynamic roles is the same** | **Immagine che contiene testo, schermata, Carattere  Il contenuto generato dall'IA potrebbe non essere corretto.** |
| **4** | **By hovering over each of the individual roles, 4 different icons appear.**  **which respectively allow**   * **Show all code references to that element** * **Show the part of code related to the definition of the element** * **Allows editing of the name** * **Deletion of the item (upon confirmation)** | **Immagine che contiene testo, Carattere, schermata  Il contenuto generato dall'IA potrebbe non essere corretto.**  **Immagine che contiene testo, schermata, Carattere  Il contenuto generato dall'IA potrebbe non essere corretto.**  **Immagine che contiene testo, schermata, Carattere  Il contenuto generato dall'IA potrebbe non essere corretto.**  **Immagine che contiene testo, Carattere, schermata  Il contenuto generato dall'IA potrebbe non essere corretto.** |

## Thumbnail Image Crea un'immagine di dipendenti in piedi che guardano in camera con lo sfondo di un ufficio open space. L'ufficio dovrebbe avere un design moderno con ampie finestre, scrivanie ordinate e attrezzature tecnologiche. I dipendenti devono essere vestiti in abbigliamento professionale, mostrando diversità in termini di età, genere e etnia. L'atmosfera dovrebbe essere luminosa e accogliente, con colori neutri e dettagli in verde per rappresentare un ambiente di lavoro produttivo e collaborativo.SandBox Data: Users

In order to run the SandBox locally and thus be able to test the flow, it is necessary to create the example data in the form of a Json file.

Immagine che contiene testo, schermata, menu

Il contenuto generato dall'IA potrebbe non essere corretto.The first file to be edited is the **users.json** whose contents must be consistent with the static Roles defined and the attributes that a user will need to have for the workflow logic to work properly.

Next you see an example of elements.

Standard keys

* **UserName:** Unique identifier of the user.
* FullName**:** Name of the user that is shown on the interface
* Roles: Roles impersonated by the user separated by comma
* CompanyName: Company name that is shown in the interface;
* **Email**: Email of the user being used by the code.

Custom Keys:

Under the **Profile** key you can add other attributes that can then be read at runtime. For our example we will create the **ManagedBy** key with the UserName of its direct user.

## Thumbnail Image Crea un'immagine che rappresenti tabelle di dati di un database relazionate fra loro. Mostra diverse tabelle con colonne e righe, con linee che collegano le chiavi primarie e le chiavi esterne tra le tabelle. Utilizza uno stile professionale e pulito, con colori neutri come il blu e il grigio. Includi etichette chiare per ogni tabella e colonna, e mostra esempi di dati realistici. L'immagine dovrebbe trasmettere un senso di organizzazione e struttura.Business Object

As the name suggests, Business Objects (BOs) represent real-world entities used in the context of data management and business processes. These entities can be customers, products, orders, and many other figures essential to the operation of a business.

Business Objects are data models that define the characteristics and properties of business entities. Each BO has a set of attributes that describe its properties and may vary depending on the nature of the entity. For example, a Customer BO might include attributes such as name, address, phone number, and email.

### Business Object Examples

Client: This BO could have attributes such as:

* Customer ID (unique)
* Client name
* Address
* Phone number
* Email

Product: This BO might include:

* Product ID (unique)
* Product name
* Category
* Price
* Stock available

Order: The attributes of this BO could be:

* Order ID (unique)
* Customer ID
* Product ID
* Order date
* Quantity
* Order Status.

For the Flight workflow we identify three different BOs:

Airline: Airline Directory. Used in airline choice forms. WorkItem references an instance of it.

|  |  |  |  |
| --- | --- | --- | --- |
| Airline | | | |
| ID | Int | 1,2,3,4,5 |
| Code | String | "AA," "BA," "DL" |
| Name | String | "American Airlines," "British Airlines," "Delta." |

Airport: Airport Directory. Used in forms to select the departure and return airport. Workitem references two instances of this.

|  |  |  |
| --- | --- | --- |
| Airport | | |
| ID | Int | 1,2,3,4,5 |
| Code | String | "CTA", "FCO", "LIN" |
| Name | String | "Catania Fontanarossa," "Rome Fiumicino," "Milan Linate." |

**CostCenter: List of cost centers for ticket cost allocation. Each user belongs to a CostCenter. The WorkItem references one instance of it.**

|  |  |  |
| --- | --- | --- |
| CostCenter | | |
| ID | Int | 1,2,3,4,5 |
| Code | String | "HR", "MK", "FN" |
| Name | String | "Human Resources," "Marketing," "Finance." |

### ****Business Object Modeling****

**Immagine che contiene Personaggio immaginario, Cartoni animati, illustrazione, Animazione

Il contenuto generato dall'IA potrebbe non essere corretto.**

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Example** |
| **1** | **Position yourself on the YS tree and select the "Business Objects" branch.** |  |
| **2** | **You then click on the icon and in the top bar you are asked to name it.**  **Enter the name Airline**  **The new BO will be created automatically, which includes:**   * **An attribute list where the main one is ID which gives us the uniqueness** * **One or more Forms representing us individual BO** |  |
| **4** | **By hovering over the BO name, 4 different icons appear.**  **which respectively allow**   * **Show all code references to that element** * **Inserts a new WorkItem attribute.** * **Allows editing of the name** * **Deletion of the item (upon confirmation)** |  |
| **5** | **Let us now create the remaining attributes of the BO Airline. That is, Name and Code of type string.**  **Clicking on the icon brings up a list of all the possible types of attributes you can create.**  **We will see them later when we talk about the WorkItem. Limit yourself now to choosing the type "string"** | Immagine che contiene testo, schermata, software, Software multimediale  Il contenuto generato dall'IA potrebbe non essere corretto. |
| **6** | **Once the task is completed, the result should look as in the figure.**  **Note that if you are in Developer mode, auto-generated C# code also appears at the side.**  **Handle with caution!** |  |
| **7** | **We complete the business by also creating BO Airport and CostCenter** |  |

### **Immagine che contiene testo, schermata, software Il contenuto generato dall'IA potrebbe non essere corretto.**C# coding of Business Objects

**Once the Business Object is defined, one can move on to the C# coding phase in order to handle more technical aspects of their operation.**

**A newly generated BO is shown at the side. In the first line identify the decoration DbBo this will indicate that the BO is a database type. YS currently supports only this type of BO while others will be added later (Web Service, InMemory, Custom)**

**The possible attributes of DbBo are as follows.**

[DbBo(DescriptionProperties= new string[] { "Code", "Name" },ConnectionName= "ExternalDB", TableName= "AIRLINE", IdPropName= "AL\_ID")]

**Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.DescriptionProperties**: is of string array type and indicates the various attributes that make up the descriptive part of the element. This is typically used in Form drop-down lists, providing a clear and concise description of the selectable elements.

**ConnectionName**: is of type string and specifies the name of the database connection used. This field is critical to ensure that the BO can properly access the data stored in the database once in production.

"dBConnections": [

{

"name": "YubikDB",

"connectionString": ""

},

{

}, "name": "ExternalDB",

"connectionString": " Server=localhost;Database= ExternalDB;User Id=sa;Password= your\_password;"

}

],

**TableName**: is of type string and defines the name of the database table associated with the BO. Through this attribute, the BO knows where to find the data it needs to manipulate.

**IdPropName**: is of string type and identifies the unique field of the database table. This field is used to ensure the uniqueness of elements within the table, facilitating search and data management operations.

At the moment we are only going to define the attribute "DescriptionProperties" with the fields "Code" and Name"

## Business Object (EnumValue)

[Flags]

public enum Class

{

Ecomomy= 1,

Business= 2,

First= 3,

Premium= 4,

}

There is also a simpler type of BO consisting only of a list of fixed values such as, for example, the class of a flight (Economy, Business, ...). This structure is very convenient to use as it is linked to drop-down lists in Forms.

### ****Business Object Modeling (EnumValue)****

**Immagine che contiene Personaggio immaginario, Cartoni animati, illustrazione, Animazione

Il contenuto generato dall'IA potrebbe non essere corretto.**

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Example** |
| **1** | **Position on the YS tree and select the "Business Objects" branch🡪 "EnumValue"** |  |
| **2** | **You then click on the icon and in the top bar you are asked to name it.**  **Enter the name Class**  **The new BO of type EnumValue empty will be created automatically** |  |
| **4** | **Now enter the list of text values by giving them an identifying number as in the example shown** |  |

## Immagine che contiene testo, Carattere, schermata, design Il contenuto generato dall'IA potrebbe non essere corretto.SandBox Data: Business Objects

Once the different BOs of the solution have been defined, the relevant JSON files that will persist the data can be created within the SendboxData/BO section. To do this, a set of well-defined steps must be followed:

Immagine che contiene testo, schermata, menu, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.Each of the JSON files must be an array of objects. These objects must have all the attributes defined in the BO. These attributes represent the properties and characteristics specific to the BO, such as name, type, and other relevant details.

In addition to the BO-specific attributes, an additional attribute called "BusinessObjectType" must be added. This attribute will have as its value the name of the related BO. This is critical because it allows YS to easily identify the BO type to which each object belongs within the JSON file.

The JSON files created will be used for data persistence throughout the early stages of development without having to access external resources such as Database servers

### ****SandBoxData/Business Object Modeling.****

**Immagine che contiene Personaggio immaginario, Cartoni animati, illustrazione, Animazione

Il contenuto generato dall'IA potrebbe non essere corretto.**

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Example** |
| **1** | **Place on the YS tree and select the "SandBoxData" branch🡪 "BO"** |  |
| **2** | **Then click on the icon , and the top bar asks for the name to be given to it.**  **Enter the name Airline**  **The new data file of the empty BO will be created automatically** | **(Function not yet available)** |
| **4** | **An array of Json objects must be created for inserting the BO elements. Each element will contain as keys all the defined BO fields and as the last key should be inserted "BusinessObjectType" : "Airline"** |  |

## Workitem

The Workitem constitutes one of the fundamental structures of a solution, as it includes all relevant properties related to the practice. Each Workitem represents a process instance that aggregates relevant data and documents, enabling efficient and integrated management of activities.

**Immagine che contiene testo, schermata, software, Software multimediale

Il contenuto generato dall'IA potrebbe non essere corretto.**A Workitem is composed of different types of properties:

* **Base Types**: Standard Single Fields
* String, Integer, Decimal, DateTime, DateOnly, TimeOnly, Boolean. All in the "nullable" exception or not
* **Core Typed**: Fields of system structures
* Identifier of a document (Document)
* Support attribute of a dynamic role (DynamicRole)
* Identifier of a user (User)
* **Business Objects**:
* Single instances of a specific BO created (Airline
* Single instances si an Enum (Class)
* Collections of instances of a specific BO (iCollection< Airline>)

### ****WorkItem modelingImmagine che contiene Personaggio immaginario, Cartoni animati, illustrazione, Animazione Il contenuto generato dall'IA potrebbe non essere corretto.****

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Example** |
| **1** | **Stand on the YS tree and select the "WorkItem" branch of the Flight project.** |  |
| **2** | **Then click on the icon, and in the top bar you are asked to select the type of attribute to create from those listed. For example string**  **Once the type is selected, it is asked to define its name, which must be unique.** |  |
| **4** | **Repeat the operation for all the properties to be created. Eventually, you should get a list as in the figure** |  |

### **Immagine che contiene testo, schermata, software, Sistema operativo Il contenuto generato dall'IA potrebbe non essere corretto.**WorkItem c# coding

**Once the Business Object is defined, one can move on to the C# coding phase in order to handle more technical aspects of its operation.**

**Basically what is required is just to specify the size in characters of the string fields.**

**In the example at the side we set the string "RequestNote" to 1000 characters and initialized some fields.**

**It is also possible to initialize the various WorkItem fields with an initial value.**

## Process Map

A possible STAR representation of the process is shown below. The student is asked to model the same process with YS using the same Stage and Action names.

In order to model the Map most effectively, we recommend that you open the Yubiq Studio tree in the way you see so that you can easily drag and drop the elements you see with the white rectangles within the map.

If you position yourself in the lower right corner, in the bar you will find the option to switch from "Modeller" mode To "Developer" mode.

Immagine che contiene schermata, testo, diagramma, Software per la grafica

Il contenuto generato dall'IA potrebbe non essere corretto.

Below is a table to guide you in assigning different roles to Internships.

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Static Roles | Dynamic Roles | Motivation |
| CreationAction: FlightRequest | Employee, Director |  | Allows the creation of a new WorkItem to the indicated roles. Since each user has at least one of two roles, anyone can request an airline ticket |
| Internship: MyRequests |  | Traveler | The dynamic Traveler role is assigned to the WI same time as its with the previous action. so we indicate that each Traveler will access its own requests present in the Stage.  Since the 4 actions that depart from this Stage are all executable by those who have access to this stage we do not need to specify roles but only indicate that execution is allowed to those who have WI in ToDo list |
| Internship: OnHold  Internship: MyTakenflight |  | Traveler | In these Stages the Traveler accesses their own requests. |
| Internship: RequestApproval |  | Manager | The dynamic Manager role is also assigned to the WI upon its creation. So each Manager access this Stage only for their own requests to be approved |
| Internship: TicketBooking | TravelDept |  | Only users with the TravelDept role have access to this stage. |
|  |  |  |  |

Note: All Stages and Actions are always accessed by those who hold the static system role "Admin"

### ****Modeling the Process****

**Immagine che contiene Personaggio immaginario, Cartoni animati, illustrazione, Animazione

Il contenuto generato dall'IA potrebbe non essere corretto.**

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Example** |
| **1** | **Position yourself on the YS tree and select the "Workflows" branch and then "Flight"** |  |
| **2** | **Then click on the icon to show the initial map and then proceed with modeling** | Immagine che contiene schermata, testo, diagramma, Software per la grafica  Il contenuto generato dall'IA potrebbe non essere corretto. |
| **3** | **In order to create the stages, one must be located on Process, click on the icon and enter a unique Stage name. It will then appear on the Map** |  |
| **4** | **Actions are created directly in the Map. This is done by selecting the starting Stage with a click and then clicking on the ending Stage while holding down the "Shift" key**  **You are then asked for the name of the Action and The type of Action among those available.**  **If you choose the User type action you are asked for the Form to be shown from those available.**  **If you have not yet created the desired Form we recommend temporarily choosing the one that is created by default when creating the FlightDisplay Workflow** |  |
| **5** | **It is also possible to change the destination of an Action. This is done by clicking on an Action so that the destination arrow appears dashed, then holding down "Shift" are click on the desired destination stage.** |  |
| **6** | **In order to add the desired roles to stages and actions you can drag and drop roles from the tree to the Map.**  **Select the Role "Director" drag the element to the Stage or Action**  **Then hold down the "Shift" key and release it when you have reached the Stage or Action with which to associate it.** |  |
| **7** | **Similarly, it is possible to associate Forms with Stages or Action** |  |

### Process c# coding

Once the process has been modeled and user roles appropriately assigned, we move on to writing code in the various Stages and Actions.

**Internship:**

Immagine che contiene testo, schermata, Carattere, linea

Il contenuto generato dall'IA potrebbe non essere corretto.A Stage intercepts two specific events where to enter the code:

**OnEnter**: When the WorkItem lands in the Stage coming from an Action connected to it.

Immagine che contiene testo, Carattere, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.**OnExit**: When the WorkItem restarts from the Stage with an Action attached to it.

These two events must be explicitly created by clicking "+" if you want to insert c# code. Pressing "+" creates the method in the appropriate class where to insert, assisted by AI the code.

**Action:**

An Action also intercepts events but these also depend on the type of Action.

**User Action:**

**Immagine che contiene testo, Carattere, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.**A User Action intercepts three specific events where to insert code:

Immagine che contiene testo, Carattere, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.**CanView**: YS calls this method before showing the Action among those available to the user so as to execute some code that can return True or False and then show the Action accordingly

**OnPrepare**: YS calls this method in order to prepare the Form assigned to the Action.

**Immagine che contiene testo, Carattere, schermata, Elementi grafici

Il contenuto generato dall'IA potrebbe non essere corretto.OnExecute**: YS calls this method the moment the user has given the Form Confirmation in order to execute the related code

**Conditional Action:**

### Forms

Immagine che contiene testo, schermata, schermo, numero

Descrizione generata automaticamente

**Flightrequest**: Form for requesting and modifying requested ticket information. In ReadOnly mode reports the request information. You will need to request the applicant's ID card

Immagine che contiene testo, schermata, schermo, numero

Descrizione generata automaticamente

**FlightReservation**: Form for entering and editing ticket information. In ReadOnly mode reports the ticket information. You will need to request a photo of the airline ticket

Immagine che contiene testo, schermata, Rettangolo, schermo

Descrizione generata automaticamente

**VPApprouval**: Response form in VicePresident's approval.

Immagine che contiene testo, schermata, schermo, numero

Descrizione generata automaticamente

**Flightrequest**: Form for requesting and modifying requested ticket information. In ReadOnly mode reports the request information. You will need to request the applicant's ID card

Immagine che contiene testo, schermata, schermo, numero

Descrizione generata automaticamente

**FlightReservation**: Form for entering and editing ticket information. In ReadOnly mode reports the ticket information. You will need to request a photo of the airline ticket

Immagine che contiene testo, schermata, Rettangolo, schermo

Descrizione generata automaticamente

**VPApprouval**: Response form in VicePresident's approval.