**Service Studio:**

Service Studio is the visual development environment for OutSystems Platform, used to design web and mobile applications including their user interfaces, process flows, business logic, web services, and security policies.

**Controls:**

Web server controls are objects on Web pages that run when the page is requested and that render markup to a browser. Many Web server controls resemble familiar HTML elements, such as buttons and text boxes. Other controls encompass complex behavior, such as a calendar controls, and controls that manage data connections like List records or table records.

Container control

Place holder

Expression control

If condition control

List records or table records :

Table Records doesn't provide a flexible layout, since it’s mainly used to display the data in a table based layout.

List Records a very flexible and customizable layout., since it’s mainly used to display the data in any layout. Provides facility to insert another control inside it but table records doesn’t.

Editable Table: Allow to edit a single/multiple record in tabular format.

Edit Record: Allow editing a single record in tabular format.

Show Record:  Shows a single record in tabular format.

Text box

Check box

Radio button

Combo box: A dropdown list to select single value from list

List box: A dropdown list to select multiple values from list.

Buttton

Components:

There are more than 500 components found for Web application, some are below:

Silk UI WEB

Google Map

Location

Advance Excel

Html 2pdf converter

Excel packages

PayPal connector

Date Picker

Widgets

Popup

Input autocomplete :

Place a textbox with input\_autocomplete.

Place an action for autocomplete widgets.

Here get all items on behalf of search text that needed to bind with “Input\_AutoComplete\_ShowList” action list widget.

[Feedback Ajax Wait](https://andrefmota.outsystemscloud.com/RichWidgetsSample/FeedbackAjaxWait.aspx?%28Not.Licensed.For.Production%29=)

Shows a message if Ajax calls take too much time to show.

[Input Set Focus](https://andrefmota.outsystemscloud.com/RichWidgetsSample/InputSetFocus.aspx?%28Not.Licensed.For.Production%29=)

Sets the focus on the specified Input or Input Password widget when the screen loads.

[Popup Upload](https://andrefmota.outsystemscloud.com/RichWidgetsSample/PopupUpload.aspx?%28Not.Licensed.For.Production%29=)

A page to be used in a Popup\_Editor to upload files.

[Remove Popups](https://andrefmota.outsystemscloud.com/RichWidgetsSample/RemovePopups.aspx?%28Not.Licensed.For.Production%29=)

Prevents the page from appearing inside a popup.

Place an action for textbox change at textbox destination , in which assign the selected identifier into a variable using autocomplete action “Input\_AutoComplete\_GetIdentifier”

Popup\_InfoBalloon

FakeNotifyWidget

Tabs\_ClientSide

Extension : is a simple way to create our own code with .net/java

**Integration studio**

Integration studio providing such facility to create and compile extainsion on specific server. After compilation and publishing of extension we add this extension to our page by adding reference.

**Integration**

With OutSystems Platform, integration is easy. Developers manage the configurations of integration without the need to write custom code, significantly reducing time and effort and eliminating errors. Integration with SOAP and REST services and SAP systems is built in, and OutSystems Platform generates all the methods and data structures to integrate with the external system. Developers can then use them visually in the application logic. From a developer's perspective, there is almost no difference between invoking an OutSystems method, a REST API, or an SAP BAPI.

API *OutSystems Platform provides you APIs that you can use to extend the capabilities of your applications. With these APIs you can integrate your applications with external systems, and have programmatically access to the features of the platform.*

Crypto API ->(cryptographic lib Providing authentication encryption )compare password,compute hash ,encrpt and decrpt

'DBCleaner\_API' ->For Platform Version 9.0.1.4. Allows freeing-up database space.

[Charts API](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Overview_of_the_Charts_API.htm)à> The OutSystems Platform API for plotting charts.

          To use this API simply reference the Charts eSpace using the [References Window](https://www.outsystems.com/help/servicestudio/9.0/Reusing_eSpaces/Add_Remove_References_Window.htm) in the development environment.

**Chart Widgets:**

     The charts available to use are:

[AreaChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/AreaChart_Web_Block.htm)

[BarChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/BarChart_Web_Block.htm)

[ColumnChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/ColumnChart_Web_Block.htm)

[LineChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/LineChart_Web_Block.htm)

[PieChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/PieChart_Web_Block.htm)

         Life Time API –

               With LifeTime you have full visibility of all applications across all environments in your      infrastructure.

REST

It is simple Stateless architecture tht generally run over Http

      REpresentational State Transfer

Determined to make the integration with REST APIs as simple as possible, OutSystems has introduced an entirely new way to automate the consumption of a REST API.

There is no contract in REST APIs, Simply typing the address for the REST API will enable OutSystems Platform to automatically discover the underlying structure present. It then creates an interface that can be dragged around in the visual logic as if it were any other piece of business logic or web service. Users just need to fill in the parameters and start using the API.

OutSystems Platform creates all the required structure to support impact and change analysis throughout the app. If the API changes, everything is updated, automatically healed and checked for impact

To consume a REST API, a developer needs to provide the service end-point URL, and an example of the service request and response in JSON. Most services include a JSON sample request and response in their documentation and it can be pasted into the OutSystems Platform IDE. OutSystems Platform generates everything needed to invoke the REST service.. /PhoenixBilling/rest/RESTAPI1

Web service - > to Access data frm different sources like web software app

SOAP

      Simple object  Access Protcal based on XML so it easy to read

                 To consume a SOAP web service, developers simply provide the location of the     WSDL (Web Services Description Language) file. OutSystems Platform IDE inspects the WSDL and generates everything needed to invoke the web service methods.

<https://172.16.32.28/Login/GlobalUsers.asmx?wsdl>

Exposing SOAP web services is done by exposing the methods. OutSystems Platform will automatically create the WSDL and URL that can be consumed by any application with Internet connectivity.

SSL (Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral.

Timer

To execute asynchronous logic, use Timers. This is useful to execute batch tasks like sending emails at a predetermined time, or to execute logic to configure an application after its deployment.

A timer can be awaked in three different ways:

On a predetermined time;

After the application has been published;

Either programmatically or manually.

**Processes**

A Process is an element that allows you to integrate your business processes into your applications. A process is designed in a process flow, which usually represents the activities that have to be carried out during an entity life cycle.

Elements

**Entitiy**

An entity is an element which allows you to keep business information in a persistent way. The information is stored in Attributes. Entities are used to represent and manage your data base model.

**Static Entities**

A static entity is an entity that has static data associated to it. This static data is then managed in design time and you use it directly in the business logic design of your application thus benefiting from strong typing. You can think of Static Entities as enums that are stored in the data base

**Multi-tenant**

To create a multi-tenant application with OutSystems Platform, every eSpace of the application has to be marked as multi-tenant. This is done by simply setting the "Is Multi-tenant" property of the eSpace to "Yes." This isolates data, end-users, sessions and processes per tenant and alerts other developers to the fact that the eSpace is multi-tenant ready.

Since an end-user uniquely belongs to a tenant, the OutSystems Platform infers which tenant from the username, during the login process. Then, the end-user is bound to the tenant and all simple and advanced queries are automatically filtered to only return data from that tenant

**Steps of upload and download release**

First tack all the dependent eSpace modules references in our module Open service center and go to factory menu then go to Solutions sub menu then click on new Solutions link

Click Publish Steps

Save (optional): this is an optional step that is performed only if your module was opened from a file or was already saved to a file.

Upload: uploads the module to the Platform Server you are connected to.

Compile: this is executed in the Platform Server you are connected to and involves translating the oml file, handled by the development environment, into a set of Microsoft .NET or Java files.

Deploy: the last step of the 1-Click Publish operation which updates the published version of the module.

LifeTime

LifeTime fully automates DevOps processes, and manages the staging of your applications from development to production across cloud, on-premises or hybrid environments.

Real-time performance monitoring is built-in for all your applications and available in LifeTime. You’ll be able to see how each web and mobile application is performing on your users' devices.

LifeTime also enables you to define your IT team's access and control over applications and their components, across all environments.

**Service Center**

Service Center is OutSystems Platform web console that offers a comprehensive set of functionality to manage all operational aspects of your applications, including all logging and monitoring information needed for performance tuning, and for configuration of horizontal scalability.

**What is Agile?**

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.

It works by breaking projects down into little bits of user functionality called [user stories](https://www.agilenutshell.com/user_stories), prioritizing them, and then continuously delivering them in short two week cycles called [iterations](https://www.agilenutshell.com/iterations)/Sprints.

Deployment process:

SAP Integration

Site property-One of the benefits of using site properties, is that you can change its value at runtime using the environment management console.

Outsystem 9 bali : Analytics: OutSystems Platform Bali takes mobile and web application performance management to a new level by providing innovative, holistic, analytics across the application portfolio.:

Model

The default value for the session time-out depends on the application server:

.NET: 20 minutes and this value can be configured in machine.config file;

J2EE: 60 minutes and this value can be configured in /etc/.java/.systemPrefs/outsystems/prefs.xml file

A vendor management system (VMS) is a Web-based application that allows an organization to secure and manage staffing services on a temporary, permanent or contract basis. It helps centralize the complex issues that surround the staffing.

A VMS generally involves the following:

Job requisition or staff ordering

Automatic billing

Business intelligence (BI) functionality

Management reporting

Workflow engines

[Getting Help](https://success.outsystems.com/Support)

[Enterprise Customers](https://success.outsystems.com/Support/Enterprise_Customers)

[Maintenance and Operations](https://success.outsystems.com/Support/Enterprise_Customers/Maintenance_and_Operations)

[Designing the architecture of your OutSystems applications](https://success.outsystems.com/Support/Enterprise_Customers/Maintenance_and_Operations/Designing_the_architecture_of_your_OutSystems_applications)

The 4 Layer Canvas (4LC) is an OutSystems architecture tool to make the design of Service-Oriented Architectures (SOA) simple. It promotes the correct abstraction of reusable (micro)services and the correct isolation of distinct functional modules, in cases where you are developing and maintaining multiple applications that reuse common modules. A typical medium to large OutSystems installation will support 20+ mission critical applications and 200+ interdependent modules.

**The 4 layers**

Each layer sets a different nature of the functionality to be captured in a module:

|  |  |
| --- | --- |
|  | Processes, dashboards and portal home pages, mashing up information from different applications to provide a unified user experience. |
|  | User interfaces and processes, reusing **Core**and **Library**services to implement the user stories. |
|  | Services around business concepts, exporting reusable entities, business rules and business widgets. |
|  | Business-agnostic services to extend the framework with highly reusable assets, UI Patterns, connectors to external systems and integration of native code. |

**Aggregates and Advanced Queries**

Use Aggregates: they’re optimized and database independent!

Use Advanced Queries for bulk operations

Use a 3 layers approach:

Business Processes: a service available for use by users or business processes

Core Entities: logical grouping of operations per responsibility

Connectors: extensions or integration with other systems

Create team processes for active log monitoring

Create team processes for manual deployment procedures such as data update scripts

If the number of lines to be uploaded is not fixed, then you should ensure that your application is capable of loading 65536 records, the maximum number of records Excel allows (Note: MS Office / Excel 2007 has a limit of 1 Million rows). Additionally, in a worst case scenario, Excel doesn't take more than two minutes to load, keeping in mind that table inserts will be progressively slower with table growth.

About Session Variables

Session variables hold data that is persisted during the [session](https://www.outsystems.com/help/servicestudio/8.0/Language_Reference/About_Sessions_in_OutSystems_Platform.htm) and can be used to save information during the end-user interaction. Each application has several session variables automatically created, but you can define new ones.

**Session Variable lifetime**

The session variables are initiated automatically by Service Studio when the [Platform Server session](https://www.outsystems.com/help/servicestudio/8.0/Language_Reference/About_Sessions_in_OutSystems_Platform.htm) is created. While the session exists, you can use these variables in your business logic. When the session ends, the session variables are set to their default value.

**Pre-defined Session Variables**

Each application has several session variables that are automatically created and read-only:

**ExceptionMessage**: Text that explains the reason for the last error, such as "Attempted to divide by zero". If the exception was raised due to an explicit Raise error, it will contain the text defined by the developer. Learn more [about exception logging](https://www.outsystems.com/help/servicestudio/8.0/Managing_Exceptions/Exceptions_Logs.htm).

Read only.

**MobilePhone**: In SMS flows, this contains the value of the phone number. In web flows this variable is also available but has an empty value.

Read only. [Phone number](https://www.outsystems.com/help/servicestudio/8.0/Language_Reference/Data_Types/Phone_Number_Data_Type.htm) type.

**UserId**: Contains the unique identifier of the end-user making the request. This variable is instantiated during the login operation, whether you are using an explicit or implicit login. At logout, this session variable is assigned a Null Identifier value. Learn more about [Authenticating End-users](https://www.outsystems.com/help/servicestudio/8.0/Handling_security/Authenticating_End_Users.htm).

Read only.

**Username**: Contains the username of the end-user making the request. This variable is instantiated during the login operation, whether you are using an explicit or implicit login. At logout, this session variable is assigned an empty text value. Learn more about [Authenticating End-users](https://www.outsystems.com/help/servicestudio/8.0/Handling_security/Authenticating_End_Users.htm).

Read only. [Text](https://www.outsystems.com/help/servicestudio/8.0/Language_Reference/Data_Types/Text_Data_Type.htm) type.

**TerminalType**: Indicates the type of terminal that is being used to make the request. The possible values are: WAP, SMS, and Web. These values are instantiated when the session starts.

Read only.

**User-defined Session Variables**

You can define your own session variables. To define a session variable, simply right-click on the **Session Variables** folder (in the eSpace tree under the Data layer), and select the **Add Session Variable** option. Session variables, just like screen variables have an associated data type and a default value that can be set through the session variable properties. Learn more about [session variables data types](https://www.outsystems.com/help/servicestudio/8.0/Language_Reference/Data_Types/Type_of_Session_Variable.htm).

Session variables should be used with care since they can affect the scalability of your applications. You should try to avoid storing large quantities of data with session variables because, in each request, these variables need to be fetched from the database and then, when the request ends, updated in the database. When Service Studio detects you are compromising the application's performance and scalability, a [warning](https://www.outsystems.com/help/servicestudio/8.0/Errors_and_warnings/Scalability_Suggestion_Warning.htm) message is displayed.

When the end-user logs in or out, using the [Login](https://www.outsystems.com/help/servicestudio/8.0/Language_Reference/System_Actions_and_Functions/Login_Action.htm), [LoginPassword](https://www.outsystems.com/help/servicestudio/8.0/Language_Reference/System_Actions_and_Functions/LoginPassword_Action.htm), or [Logout](https://www.outsystems.com/help/servicestudio/8.0/Language_Reference/System_Actions_and_Functions/Logout_Action.htm) actions, the session variables are set to their default value.

**What you Should Know**

Asynchronous logic such as [sending Emails](https://www.outsystems.com/help/servicestudio/8.0/Web_User_Interface/Sending_EMails.htm), [Timers execution](https://www.outsystems.com/help/servicestudio/8.0/Using_Timers/Use_Timers.htm), [Processes execution](https://www.outsystems.com/help/servicestudio/8.0/Designing_Processes/About_Processes.htm), and [Web Services](https://www.outsystems.com/help/servicestudio/8.0/Integrating_with_other_systems/Web_Services/About_Web_Services.htm) run on a different session, meaning that all session variables are set with their default value when the logic is executed.

**System Roles and Custom Roles**

When you create a new module in Service Studio, OutSystems provides you with a default set of **System Roles** but you are allowed to define your own **custom Roles**.

OutSystems provides the following System Roles:

**Anonymous:** Allows any end-user to access the element, including users that are not logged in (non-authenticated users). Anonymous is the most general Role and when you associate this Role, for example with a screen, all of the existing Roles are automatically associated with it.

**Registered:** Allows any end-user who has logged into an Application running in the same platform server (authenticated users) to access the element. This is possible due to the Single Sign-On mechanism of OutSystems, which allows sharing end-user sessions among applications/modules. When you associate this Role with an element all of the existing Roles are automatically associated with it, except the Anonymous role.

Besides the System Roles already provided, you can define your own custom Roles to manage the access of end-users to the screens and operation of your application.

The following Role is provided by default when you create the first module of your application:

<Module Name>User

**Persistency in Roles**

Permissions can be persistent across multiple sessions, or only be granted for a single session.

**Persistent:** The association between the end-user and the Roles is stored in the database. Every time the end-user logs in, the association between the end-user and the Roles is established. Set the **Is Persistent** property of the Role to Yes.

**Not persistent:** The Role is only associated with the user for a single session, and not persistent in the database. When the end-user logs in for the second time, the Role is not associated with the end-user. Set the **Is Persistent** property of the Role to NO

1. Service Studio: --
2. Integration studio
3. Controls --
4. Components : --
5. Widgets --
6. Extension --
7. Integration studio --
8. Integration --
9. Web-Services or api’s : REST/SOAP (WSDL: Web service description Language ) --
10. Timers –
11. Processes --
12. Aggregate SQL and Advance Sql
13. Entities --
14. Emails
15. Implementation of js/css/themes
16. Data base integration
17. Multi tenancy --
18. Steps of upload and download release
19. Release steps
20. About life time --
21. About service center --
22. Click Publish Steps –
23. user action and screen action
24. Service Studio

Service Studio is the visual development environment for OutSystems Platform, used to design web and mobile applications including their user interfaces, process flows, business logic, web services, and security policies.

1. Controls:

Web server controls are objects on Web pages that run when the page is requested and that render markup to a browser. Many Web server controls resemble familiar HTML elements, such as buttons and text boxes. Other controls encompass complex behavior, such as a calendar controls, and controls that manage data connections like List records or table records.

* 1. Container control
  2. Place holder
  3. Expression control
  4. If condition control
  5. List records or table records :

Table Records doesn't provide a flexible layout, since it’s mainly used to display the data in a table based layout.

List Records a very flexible and customizable layout., since it’s mainly used to display the data in any layout. Provides facility to insert another control inside it but table records doesn’t.

* 1. Editable Table: Allow to edit a single/multiple record in tabular format.
  2. Edit Record: Allow editing a single record in tabular format.
  3. Show Record: Shows a single record in tabular format.
  4. Text box
  5. Check box
  6. Radio button
  7. Combo box: A dropdown list to select single value from list
  8. List box: A dropdown list to select multiple values from list.
  9. Buttton

1. Components:

There are more than 500 components found for Web application, some are below:

* 1. Silk UI WEB
  2. Google Map
  3. Location
  4. Advance Excel
  5. Html 2pdf converter
  6. Excel packages
  7. PayPal connector
  8. Date Picker

1. Widgets
   1. Popup
   2. Input autocomplete :

* Place a textbox with input\_autocomplete.
* Place an action for autocomplete widgets.

Here get all items on behalf of search text that needed to bind with “Input\_AutoComplete\_ShowList” action list widget.

[Feedback Ajax Wait](https://andrefmota.outsystemscloud.com/RichWidgetsSample/FeedbackAjaxWait.aspx?%28Not.Licensed.For.Production%29=)

* Shows a message if Ajax calls take too much time to show.
* [Input Set Focus](https://andrefmota.outsystemscloud.com/RichWidgetsSample/InputSetFocus.aspx?%28Not.Licensed.For.Production%29=)
* Sets the focus on the specified Input or Input Password widget when the screen loads.
* [Popup Upload](https://andrefmota.outsystemscloud.com/RichWidgetsSample/PopupUpload.aspx?%28Not.Licensed.For.Production%29=)
* A page to be used in a Popup\_Editor to upload files.
* [Remove Popups](https://andrefmota.outsystemscloud.com/RichWidgetsSample/RemovePopups.aspx?%28Not.Licensed.For.Production%29=)
* Prevents the page from appearing inside a popup.
* Place an action for textbox change at textbox destination , in which assign the selected identifier into a variable using autocomplete action “Input\_AutoComplete\_GetIdentifier”
  1. Popup\_InfoBalloon
  2. FakeNotifyWidget
  3. Tabs\_ClientSide

1. Extension : is a simple way to create our own code with .net/java
2. Integration studio

Integration studio providing such facility to create and compile extainsion on specific server. After compilation and publishing of extension we add this extension to our page by adding reference.

1. Integration

With OutSystems Platform, integration is easy. Developers manage the configurations of integration without the need to write custom code, significantly reducing time and effort and eliminating errors. Integration with SOAP and REST services and SAP systems is built in, and OutSystems Platform generates all the methods and data structures to integrate with the external system. Developers can then use them visually in the application logic. From a developer's perspective, there is almost no difference between invoking an OutSystems method, a REST API, or an SAP BAPI.

API OutSystems Platform provides you APIs that you can use to extend the capabilities of your applications. With these APIs you can integrate your applications with external systems, and have programmatically access to the features of the platform.

Crypto API ->(cryptographic lib Providing authentication encryption )compare password,compute hash ,encrpt and decrpt

'DBCleaner\_API' ->For Platform Version 9.0.1.4. Allows freeing-up database space.

[Charts API](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Overview_of_the_Charts_API.htm)🡪> The OutSystems Platform API for plotting charts.

To use this API simply reference the Charts eSpace using the [References Window](https://www.outsystems.com/help/servicestudio/9.0/Reusing_eSpaces/Add_Remove_References_Window.htm) in the development environment.

Chart Widgets

The charts available to use are:

* [AreaChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/AreaChart_Web_Block.htm)
* [BarChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/BarChart_Web_Block.htm)
* [ColumnChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/ColumnChart_Web_Block.htm)
* [LineChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/LineChart_Web_Block.htm)
* [PieChart](https://www.outsystems.com/help/servicestudio/9.0/APIs/Charts/Chart_Widgets/PieChart_Web_Block.htm)

Life Time API –

With LifeTime you have full visibility of all applications across all environments in your infrastructure.

1. REST

It is simple Stateless architecture tht generally run over Http

REpresentational State Transfer

Determined to make the integration with REST APIs as simple as possible, OutSystems has introduced an entirely new way to automate the consumption of a REST API.

There is no contract in REST APIs, Simply typing the address for the REST API will enable OutSystems Platform to automatically discover the underlying structure present. It then creates an interface that can be dragged around in the visual logic as if it were any other piece of business logic or web service. Users just need to fill in the parameters and start using the API.

OutSystems Platform creates all the required structure to support impact and change analysis throughout the app. If the API changes, everything is updated, automatically healed and checked for impact

To consume a REST API, a developer needs to provide the service end-point URL, and an example of the service request and response in JSON. Most services include a JSON sample request and response in their documentation and it can be pasted into the OutSystems Platform IDE. OutSystems Platform generates everything needed to invoke the REST service.. /PhoenixBilling/rest/RESTAPI1

Web service - > to Access data frm different sources like web software app

1. SOAP

Simple object Access Protcal based on XML so it easy to read

To consume a SOAP web service, developers simply provide the location of the WSDL (Web Services Description Language) file. OutSystems Platform IDE inspects the WSDL and generates everything needed to invoke the web service methods.

http://172.16.32.28/Login/GlobalUsers.asmx?wsdl

Exposing SOAP web services is done by exposing the methods. OutSystems Platform will automatically create the WSDL and URL that can be consumed by any application with Internet connectivity.

SSL (Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral.

1. Timer

To execute asynchronous logic, use Timers. This is useful to execute batch tasks like sending emails at a predetermined time, or to execute logic to configure an application after its deployment.

A timer can be awaked in three different ways:

* On a predetermined time;
* After the application has been published;
* Either programmatically or manually.

1. Processes

A Process is an element that allows you to integrate your business processes into your applications. A process is designed in a process flow, which usually represents the activities that have to be carried out during an entity life cycle.

Elements

1. Start
2. HumanActivity: Activity to be done by human person.
3. Automatic activity : Execute an action flow that execute without human intraction.
4. Execute Process : execute another process as an activity of current process.
5. Wait: Hold the process until specific events occure
6. Decision: take the decision and follow only one path.
7. Send Mail: Send mail to single or multiple person.
8. End
9. Entitiy

An entity is an element which allows you to keep business information in a persistent way. The information is stored in Attributes. Entities are used to represent and manage your data base model.

1. Static Entities

A static entity is an entity that has static data associated to it. This static data is then managed in design time and you use it directly in the business logic design of your application thus benefiting from strong typing. You can think of Static Entities as enums that are stored in the data base

1. Multi-tenant

To create a multi-tenant application with OutSystems Platform, every eSpace of the application has to be marked as multi-tenant. This is done by simply setting the "Is Multi-tenant" property of the eSpace to "Yes." This isolates data, end-users, sessions and processes per tenant and alerts other developers to the fact that the eSpace is multi-tenant ready.

Since an end-user uniquely belongs to a tenant, the OutSystems Platform infers which tenant from the username, during the login process. Then, the end-user is bound to the tenant and all simple and advanced queries are automatically filtered to only return data from that tenant

1. Steps of upload and download release

First tack all the dependent eSpace modules references in our module Open service center and go to factory menu then go to Solutions sub menu then click on new Solutions link

1. Click Publish Steps
   1. Save (optional): this is an optional step that is performed only if your module was opened from a file or was already saved to a file.
   2. Upload: uploads the module to the Platform Server you are connected to.
   3. Compile: this is executed in the Platform Server you are connected to and involves translating the oml file, handled by the development environment, into a set of Microsoft .NET or Java files.
   4. Deploy: the last step of the 1-Click Publish operation which updates the published version of the module.
2. LifeTime

LifeTime fully automates DevOps processes, and manages the staging of your applications from development to production across cloud, on-premises or hybrid environments.

Real-time performance monitoring is built-in for all your applications and available in LifeTime. You’ll be able to see how each web and mobile application is performing on your users' devices.

LifeTime also enables you to define your IT team's access and control over applications and their components, across all environments.

1. Service Center

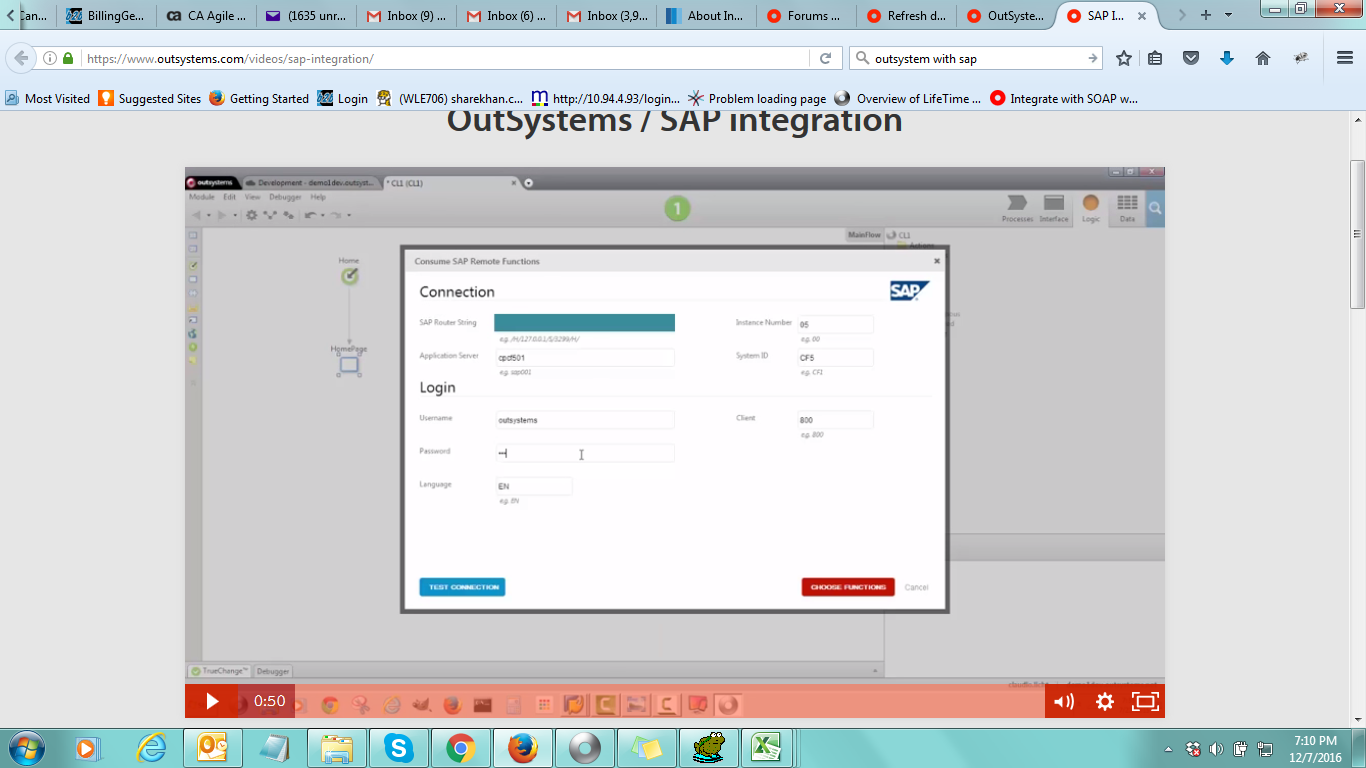
Service Center is OutSystems Platform web console that offers a comprehensive set of functionality to manage all operational aspects of your applications, including all logging and monitoring information needed for performance tuning, and for configuration of horizontal scalability.

# What is Agile?

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.

It works by breaking projects down into little bits of user functionality called [user stories](http://www.agilenutshell.com/user_stories), prioritizing them, and then continuously delivering them in short two week cycles called [iterations](http://www.agilenutshell.com/iterations)/Sprints.

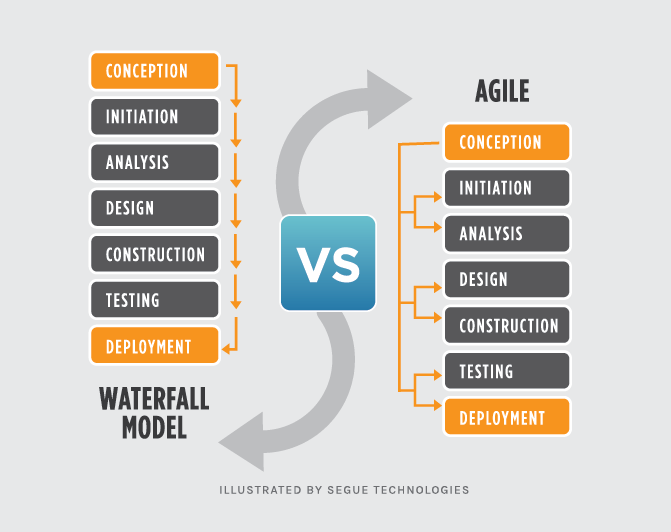
Deployment process:

SAP Integration

Site property-One of the benefits of using site properties, is that you can change its value at runtime using the environment management console.

Outsystem 9 bali : Analytics: OutSystems Platform Bali takes mobile and web application performance management to a new level by providing innovative, holistic, analytics across the application portfolio.:

Model



The default value for the session time-out depends on the application server:

* .NET: 20 minutes and this value can be configured in machine.config file;
* J2EE: 60 minutes and this value can be configured in /etc/.java/.systemPrefs/outsystems/prefs.xml file

A vendor management system (VMS) is a Web-based application that allows an organization to secure and manage staffing services on a temporary, permanent or contract basis. It helps centralize the complex issues that surround the staffing.

A VMS generally involves the following:

* Job requisition or staff ordering
* Automatic billing
* Business intelligence (BI) functionality
* Management reporting
* Workflow engines

1. [Getting Help](https://success.outsystems.com/Support)
2. [Enterprise Customers](https://success.outsystems.com/Support/Enterprise_Customers)
3. [Maintenance and Operations](https://success.outsystems.com/Support/Enterprise_Customers/Maintenance_and_Operations)
4. [Designing the architecture of your OutSystems applications](https://success.outsystems.com/Support/Enterprise_Customers/Maintenance_and_Operations/Designing_the_architecture_of_your_OutSystems_applications)

The 4 Layer Canvas (4LC) is an OutSystems architecture tool to make the design of Service-Oriented Architectures (SOA) simple. It promotes the correct abstraction of reusable (micro)services and the correct isolation of distinct functional modules, in cases where you are developing and maintaining multiple applications that reuse common modules. A typical medium to large OutSystems installation will support 20+ mission critical applications and 200+ interdependent modules.

## The 4 layers

Each layer sets a different nature of the functionality to be captured in a module:

|  |  |
| --- | --- |
| https://success.outsystems.com/@api/deki/files/2180/1__1.png?revision=1&size=bestfit&width=320&height=100 | Processes, dashboards and portal home pages, mashing up information from different applications to provide a unified user experience. |
| https://success.outsystems.com/@api/deki/files/2181/1__2.png?revision=1&size=bestfit&width=320&height=100 | User interfaces and processes, reusing **Core** and **Library** services to implement the user stories. |
| https://success.outsystems.com/@api/deki/files/2182/1__3.png?revision=1&size=bestfit&width=320&height=100 | Services around business concepts, exporting reusable entities, business rules and business widgets. |
| https://success.outsystems.com/@api/deki/files/2183/1__4.png?revision=1&size=bestfit&width=320&height=100 | Business-agnostic services to extend the framework with highly reusable assets, UI Patterns, connectors to external systems and integration of native code. |

## Aggregates and Advanced Queries

* Use Aggregates: they’re optimized and database independent!
* Use Advanced Queries for bulk operations

Use a 3 layers approach:

* Business Processes: a service available for use by users or business processes
* Core Entities: logical grouping of operations per responsibility
* Connectors: extensions or integration with other systems
* Create team processes for active log monitoring
* Create team processes for manual deployment procedures such as data update scripts

If the number of lines to be uploaded is not fixed, then you should ensure that your application is capable of loading 65536 records, the maximum number of records Excel allows (Note: MS Office / Excel 2007 has a limit of 1 Million rows). Additionally, in a worst case scenario, Excel doesn't take more than two minutes to load, keeping in mind that table inserts will be progressively slower with table growth.

About Session Variables

Session variables hold data that is persisted during the [session](http://www.outsystems.com/help/servicestudio/8.0/Language_Reference/About_Sessions_in_OutSystems_Platform.htm) and can be used to save information during the end-user interaction. Each application has several session variables automatically created, but you can define new ones.

**Session Variable lifetime**

The session variables are initiated automatically by Service Studio when the [Platform Server session](http://www.outsystems.com/help/servicestudio/8.0/Language_Reference/About_Sessions_in_OutSystems_Platform.htm) is created. While the session exists, you can use these variables in your business logic. When the session ends, the session variables are set to their default value.

**Pre-defined Session Variables**

Each application has several session variables that are automatically created and read-only:

* + **ExceptionMessage**: Text that explains the reason for the last error, such as "Attempted to divide by zero". If the exception was raised due to an explicit Raise error, it will contain the text defined by the developer. Learn more [about exception logging](http://www.outsystems.com/help/servicestudio/8.0/Managing_Exceptions/Exceptions_Logs.htm).

Read only.

* + **MobilePhone**: In SMS flows, this contains the value of the phone number. In web flows this variable is also available but has an empty value.

Read only. [Phone number](http://www.outsystems.com/help/servicestudio/8.0/Language_Reference/Data_Types/Phone_Number_Data_Type.htm) type.

* + **UserId**: Contains the unique identifier of the end-user making the request. This variable is instantiated during the login operation, whether you are using an explicit or implicit login. At logout, this session variable is assigned a Null Identifier value. Learn more about [Authenticating End-users](http://www.outsystems.com/help/servicestudio/8.0/Handling_security/Authenticating_End_Users.htm).

Read only.

* + **Username**: Contains the username of the end-user making the request. This variable is instantiated during the login operation, whether you are using an explicit or implicit login. At logout, this session variable is assigned an empty text value. Learn more about [Authenticating End-users](http://www.outsystems.com/help/servicestudio/8.0/Handling_security/Authenticating_End_Users.htm).

Read only. [Text](http://www.outsystems.com/help/servicestudio/8.0/Language_Reference/Data_Types/Text_Data_Type.htm) type.

* + **TerminalType**: Indicates the type of terminal that is being used to make the request. The possible values are: WAP, SMS, and Web. These values are instantiated when the session starts.

Read only.

**User-defined Session Variables**

You can define your own session variables. To define a session variable, simply right-click on the **Session Variables** folder (in the eSpace tree under the Data layer), and select the http://www.outsystems.com/help/servicestudio/8.0/getting_started/Images/variable.gif **Add Session Variable** option.  Session variables, just like screen variables have an associated data type and a default value that can be set through the session variable properties. Learn more about [session variables data types](http://www.outsystems.com/help/servicestudio/8.0/Language_Reference/Data_Types/Type_of_Session_Variable.htm).

Session variables should be used with care since they can affect the scalability of your applications. You should try to avoid storing large quantities of data with session variables because, in each request, these variables need to be fetched from the database and then, when the request ends, updated in the database. When Service Studio detects you are compromising the application's performance and scalability, a [warning](http://www.outsystems.com/help/servicestudio/8.0/Errors_and_warnings/Scalability_Suggestion_Warning.htm) message is displayed.

When the end-user logs in or out, using the [Login](http://www.outsystems.com/help/servicestudio/8.0/Language_Reference/System_Actions_and_Functions/Login_Action.htm), [LoginPassword](http://www.outsystems.com/help/servicestudio/8.0/Language_Reference/System_Actions_and_Functions/LoginPassword_Action.htm), or [Logout](http://www.outsystems.com/help/servicestudio/8.0/Language_Reference/System_Actions_and_Functions/Logout_Action.htm) actions, the session variables are set to their default value.

**What you Should Know**

Asynchronous logic such as [sending Emails](http://www.outsystems.com/help/servicestudio/8.0/Web_User_Interface/Sending_EMails.htm), [Timers execution](http://www.outsystems.com/help/servicestudio/8.0/Using_Timers/Use_Timers.htm), [Processes execution](http://www.outsystems.com/help/servicestudio/8.0/Designing_Processes/About_Processes.htm), and [Web Services](http://www.outsystems.com/help/servicestudio/8.0/Integrating_with_other_systems/Web_Services/About_Web_Services.htm) run on a different session, meaning that all session variables are set with their default value when the logic is executed.

## System Roles and Custom Roles

When you create a new module in Service Studio, OutSystems provides you with a default set of **System Roles** but you are allowed to define your own **custom Roles**.

OutSystems provides the following System Roles:

* **Anonymous:** Allows any end-user to access the element, including users that are not logged in (non-authenticated users). Anonymous is the most general Role and when you associate this Role, for example with a screen, all of the existing Roles are automatically associated with it.
* **Registered:** Allows any end-user who has logged into an Application running in the same platform server (authenticated users) to access the element. This is possible due to the Single Sign-On mechanism of OutSystems, which allows sharing end-user sessions among applications/modules. When you associate this Role with an element all of the existing Roles are automatically associated with it, except the Anonymous role.

Besides the System Roles already provided, you can define your own custom Roles to manage the access of end-users to the screens and operation of your application.

The following Role is provided by default when you create the first module of your application:

* <Module Name>User

## Persistency in Roles

Permissions can be persistent across multiple sessions, or only be granted for a single session.

* **Persistent:** The association between the end-user and the Roles is stored in the database. Every time the end-user logs in, the association between the end-user and the Roles is established. Set the **Is Persistent** property of the Role to Yes.
* **Not persistent:** The Role is only associated with the user for a single session, and not persistent in the database. When the end-user logs in for the second time, the Role is not associated with the end-user. Set the **Is Persistent** property of the Role to No.

## #Theme temlpates

## Dublin Template

## [Lisbon](http://www.outsystems.com/forge/component-discussions/999/Lisbon+Template)

## LiverPool

## # Good practice of Coding