Essential guide to building SaaS applications with Lianja™

Introduction

Welcome! This handy guide will help you get started by showing you what we think are the essentials for best practice when building SaaS Apps and how Lianja provides these. If you are considering Lianja and want to compare it to other products, this guide will help you.

As enterprises and small businesses (SMBs) increase their adoption of the Software as a Service (SaaS) model, developers have to consider what makes for a good SaaS solution. What elements will deliver the most return on an investment of time, effort, and money? Which ones will determine success?

There are many products out in the market that make big claims about RAD, RWAD, RMAD etc. but many lack some of the basic essentials for building and deploying SaaS solutions.

Lianja Overview

The Lianja platform is an APaaS (Application Platform As A Service). It is a highly productive, easy to learn and easy to use development and deployment environment delivering business applications that are customizable, changeable and capable of implementing serious business functionality. Deployed applications are offered with massive scalability, support for vast amounts of data, and performance and reliability that are high-end enterprise-class and beyond. All this at SMB prices.

The five pillars of Lianja are:

Lianja App Builder
Lianja App Center
Lianja Cloud Server
Lianja SQL Server
Lianja Cloud (Available soon in Q1 2019)

See also

Lianja overview

The SaaS checklist

Let's take a look at a checklist of essential SaaS development requirements and how Lianja satisfies them. When evaluating the Lianja platform for your SaaS development you can compare competing products with this list.

1. Visual native cross platform development

The IDE should be modern and provide visual drag 'n' drop of a complete application, not just be a fancy code editor. It should preferably adjust its appearance for the host operating system.

You should be able to install and run the IDE natively on your favorite operating system, develop across multiple operating systems with ease, copy databases, code and apps between different operating systems and they should just work.

See also

Lianja App Builder

	Windows	Linux	macOS
Lianja	✔	✔	✔
Other			

2. Has a built-in high performance database to reduce costs and simplify development and deployment but can connect to other SQL databases if required.

Lianja	✔
Other	

3. Live views and live debugging

To better support agile development, the IDE should provide a live view of the App as it is being developed. Compiling and building to preview a small change makes for a painfully slow development process.

	Desktop	Web	Mobile iOS	Mobile Android
Lianja	✔	✔	✔	✔
Other				

4. Live preview

The IDE should preferably provide the ability to preview Desktop, Web and Mobile Apps with one click to better support agile development.

See also

Understanding live preview in Lianja

	Desktop	Web	Mobile iOS	Mobile Android
Lianja	✔	✔	✔	✔
Other				

5. Built-in git support

Git is the most popular version control system in the world now. One of the biggest advantages of Git is its branching capabilities. Lianja keeps all the files and assets needed by an App in their own subdirectory. This makes it easy to add an App to a git repo and manage the versioning of the various files that make up the App.

See also

Using Git version Control with Lianja

Lianja	✔
Other	

6. Built-in file versioning support with the ability to roll backwards or forwards in time to see the effect of App changes during the development process.

To better support agile development of an application the IDE should allow you to make changes to the application UI and/or code, preview the changes and roll them back if required.

See also

Lianja Versions workspace

Lianja	✔
Other	

7. Cross platform App deployment

Develop native on Windows, Linux or macOS and deploy directly from the IDE to Windows, Linux, macOS, iOS or Android.

See also

Cross platform App deployment with Lianja

	Windows	Linux	macOS	iOS	Android
Lianja	✔	✔	✔	✔	✔
Other					

8. The server should run natively on Windows, Linux and macOS.

It is highly desirable that the server run natively on Linux as well as Windows as Linux is the widest used OS in AWS. This will reduce operating costs significantly. The Lianja Cloud Server runs natively on all three platforms.

See also

The Lianja Cloud Server

	Windows	Linux	macOS
Lianja	✔	✔	✔
Other			

9. Develop Desktop, Web and Mobile Apps from a single codebase using UI personalities.

Modern UI frameworks should provide attributes on the UI elements that cause them to be included/excluded for Desktop, Web, Tablet or Phone. There should be no requirement to have to code special conditions when targeting different device form factors.

Additionally, when running on mobile devices, UI elements should hide/show themselves automatically when the orientation changes from portrait/landscape. This should be able to be set during development just by setting attributes. You should not need to code this yourself.

See also

Understanding UI personalities in Lianja

Lianja	✔
Other	

10. A choice of dynamic programming language

Using dynamic scripting languages enables better agile development practices as the whole UI of the application is live during visual development. Using a dynamic scripting language rather than a non dynamic language which requires compile and build steps is highly preferable.

An added benefit of using a dynamic scripting language is that the application code can be deployed to a server and be live-updated without down time.

See also

Lianja for VFP developers
Lianja for Javascript/Typescript developers
Lianja for Python developers
Lianja for PHP developers

	LIanja/VFP	JavaScript	TypeScript	Babel/ES6	Python	PHP
Lianja	✔	✔	✔	✔	✔	✔
Other						

11. Supports NoCode development with automatic data binding

The platform should include the ability to develop Apps with no coding at all. Visually build powerful UIs using drag ‘n’ drop and adjust their appearance and functionality by setting

attributes using an App Inspector. Attribute driven development is vitally important for domain experts with little to no programming expertise.

According to Gartner, by 2020, at least 50 percent of all new business applications will be created with high-productivity toolsets such as a NoCode platform.

NoCode supports the “Design by thinking” methodology.

See also

Lianja for NoCode developers

Lianja	✔
Other	

12. Integrated database support

Apart from integrating its own native high performance database, Lianja supports transparent access to third party databases using Virtual Tables (VTs). Lianja Virtual Tables (VTs) allow you to access external data via an ODBC connection, yet they look like regular tables in a Lianja database. You can drag ‘n’ drop virtual tables in the Page Builder and reference them in the same way as standard tables and all the time the underlying connection, SQL statements and a cursor adapter are being handled for you. This makes it much easier to switch databases to conform to a corporate standard.

See also

Using Virtual Tables
Working with data

	Lianja/VFP	MSSQL	MySQL	PostgreSQL	Other ODBC
Lianja	✔	✔	✔	✔	✔
Other					

13. Has a well defined, consistent and easily understandable application architecture

Lianja has a well defined application architecture. Lianja Apps are built out of pages. Pages are built out of sections. There is a wide variety of built-in sections. Form sections for example are made up of FormItems and Gadgets. So as we can see, a Lianja App consists of a hierarchy of

visual elements. This hierarchy we will refer to as the Lianja Object Model (LOM).

You can consider the whole of the Lianja App Builder as a meta-framework that you can use to develop Apps with.

Having a well defined application architecture is critically important for NoCode development as it provides an abstraction above the target UI so that code generation is deterministic and simplified as the layout, appearance and UI navigability can be easily understood at code generation time and can target multiple form factors and target devices.

See also

The Lianja Application Architecture
Lianja UI Navigation

Lianja	✔
Other	

14. Has a modern application framework that is cross platform and cross device

If the need arises to build a custom UI the platform should have a well defined set of framework classes that can be used by professional developers in any of the supported dynamic scripting languages.

See also

Lianja Framework Classes

Lianja	✔
Other	

15. The Framework should support building responsive UIs.

Responsive design is an approach to UI creation that makes use of flexible layouts. The goal of responsive design is to build a UI that detects the user's screen size and orientation and will change the layout accordingly.

See also

Understanding UI layouts in Lianja
Understanding UI personalities in Lianja

Lianja	✔
Other	

16. Supports UI theming without needing any code changes

The platform should have the ability to theme the UI without jumping through hoops. At the bare minimum the UI should be able to be themed using industry standard CSS.

Lianja	✔
Other	

17. Built-in support for UTF-8 character encoding without code changes

UTF-8 can represent any character in the Unicode standard. UTF-8 is backwards compatible with ASCII. UTF-8 is the preferred modern character encoding. All characters of all languages are available in UTF-8. If you do not use UTF-8 in your applications your application will be locale specific. Most importantly, UTF-8 is Web and mobile friendly. It provides the ability to have characters from multiple languages on the same web pages. In other words you can integrate English, Chinese, Japanese and other languages without needing to do anything special in the UI or the database.

Lianja	✔
Other	

18. Database engine supports encrypted tables

When an application requires that certain database tables need to be encrypted to prevent them being read if for some reason security is breached, then the database engine should support the ability to encrypt and decrypt the data using encryption keys.

Lianja	✔
Other	

19. Database engine has built-in support for chronological data versioning and audit trails

Audit trails and data security go hand in hand and it is highly desirable that this functionality is built-in. Lianja database timelines provide row versioning for database tables for all CRUD operations performed on data. Whenever a change is made to a table that is timeline enabled then delta changes are automatically recorded for each transaction. Changes made to any tables that are timeline enabled can be undone much like you would undo changes to program code that you edit in a text editor.

Database timelines record who did what from where, when they did it and what they changed. This is an invaluable feature for SaaS applications.

No coding is required to be able to leverage this functionality.

See also

Understanding database timelines in Lianja

Lianja	✔
Other	

20. Built-in native JSON support

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

It is highly desirable for the native database engine to have support for JSON. Lianja takes this to an extreme level by extending SQL to include a native JSON datatype and the ability to store JSON documents and query on them (in fact full CRUD operations) using standard SQL commands without requiring the use of any special functions.

See also

Working with JSON

Lianja	✔
Other	

21. Built-in native metadata support

When building large enterprise class applications with 100s of tables that are used in multiple Apps and their associated Lianja UI elements, it can be tedious and error prone to apply the same UI attributes consistently throughout the UI.

To alleviate this problem and to speed up App development, the Lianja App Builder and the Lianja engine handle metadata. This mechanism provides a way of dynamically setting up UI element attributes when a new UI section is created or when an App, its pages and their corresponding UI elements are loaded.

Also, and probably more importantly, any changes to the database schema metadata for tables and their columns are propagated throughout Lianja Apps dynamically as they are loaded and applied anywhere they are used in the Apps and their UI elements (Sections, Fields, Grid Columns).

Let's say for example whenever the bank balance field is negative it should have a pink background and when positive it should have a light green background. This can be defined in the column metadata in a database table.

This massively speeds up application development and can be altered in one place without needing to know where the bank balance field is used in the application.

Metadata supports the "Design by thinking" methodology.

See also

The MetaData Editor

Working with metadata

Lianja	✔
Other	

22. Built-in native very large database tables support

Databases used in modern SaaS applications can quickly grow in size particularly if images and videos are stored in them. To properly support a large number of users the database should be able to grow to huge file sizes of 2^64 rather than be limited by the older 32 bit file system limits of 2GB.

Lianja	✔
Other	

23. Fault tolerant with automatic index healing

Most SaaS applications run in the Cloud. They are timezone agnostic. If for whatever reason a database index becomes corrupted, it is highly desirable that the database engine recognize this and automatically repair the index without any noticeable downtime while users from many geographically remote locations are authenticated and are running applications.

Lianja	✔
Other	

24. Built-in user authentication with permissions and roles and optional user expiry date

Having the ability to manage users is very important to maintaining security of operations and controlling access to data.

Permissions and Roles provide the ability to control CRUD access to Apps, Pages, Sections, Fields and Gadgets (the whole UI) depending on who the user is who has authenticated and what roles they have (which can be multiple).

Having the ability to specify a user expiry date for a user account simplifies SaaS App monetization. If specified, this expiry date should be checked on each page request to the server.

See also

Understanding Lianja roles and permissions

Lianja	✔
Other	

25. Built-in App dashboard with access control using permissions and roles for authenticated users

A consistent interface for logging in and then presenting the user with a categorized list of Apps that they have permission to run is important.

See also

The Lianja App Center
Customizing Lianja App Center tiles

Lianja	✔
Other	

26. Built-in row level security (RLS) for authenticated users

Row-Level Security provides the ability to control access to rows in a database table based on the characteristics of the user executing a query (e.g., role or group memberships) without having to hard code this into an application.

Row-Level Security (RLS) simplifies the design and coding of security in your application. RLS enables you to implement restrictions on data row access. For example, ensuring that workers can access only those data rows that are pertinent to their department, or restricting a customer's data access to the data relevant to their company.

See also

Understanding row level security in Lianja

Lianja	✔
Other	

27. Built-in data masking (DDM) for authenticated users

Dynamic data masking (DDM) limits sensitive data exposure by masking it to non-privileged users. It can be used to greatly simplify the design and coding of security in your application.

Dynamic data masking helps prevent unauthorized access to sensitive data by providing the ability to designate how much of the data to reveal while having minimal impact on the application layer. DDM can be configured on the database to hide sensitive data in the result sets of queries on specific columns, while the data in the database is not changed. Dynamic data masking is easy to use with existing applications, since masking rules are applied in the query results. Many applications can mask sensitive data without modifying existing queries.

See also

Understanding Dynamic Data Masking in Lianja

Lianja	✔
Other	

28. Supports cross platform LDAP/Active Directory for user permissions and roles

To satisfy corporate standards the platform should provide the ability to allow user authentication using Active Directory and/or LDAP (cross platform). The groups that the user is a member of can then be used as the user's specified roles in order to control permissions, row level security and dynamic data masks.

See also

Using Active Directory in Lianja

Lianja	✔
Other	

29. Deploy to native Desktop

Electron application code generation and building directly in the IDE is very desirable.

See also

A guide to building Lianja Electron Apps

Lianja	✔
Other	

30. Deploy to the Web

See also

A guide to deploying Web Apps

Lianja	✔
Other	

31. Deploy to native Mobile

Phonegap and/or React-native code generation and building directly in the IDE is highly desirable.

See also

A guide to building Lianja PhoneGap Apps

Lianja	✔
Other	

32. High performance and scalable

The performance of single instance servers will eventually max out. To be able to scale up and down is important to user satisfaction and long term acceptance of the SaaS application. Dynamic load balancing of server instances and shared back end storage is highly desirable. Linux and Docker containers are recommended for scalability and cost effectiveness.

Lianja Cloud	✔
Other	

33. Secure

SSL encrypted http traffic is a necessity in a production server. Lianja Cloud Server can work with Windows/IIS and Linux/Apache.

See also

Lianja IIS extension for IIS
Lianja Apache module for Linux

Lianja Cloud	✔
Other	

34. High availability and fault tolerant

Lianja Cloud	✔
Other	

35. Multi-tenant centralization without code changes

Lianja Cloud	✔
Other	

36. Live in-place database schema updates handled by the server

Lianja Cloud	✔
Other	

37. Live in-place App and code updates handled by the server

Lianja Cloud	✔
Other	

38. Has a REST architecture with an open API that works with all supported databases

OData (Open Data Protocol) is an ISO/IEC approved, OASIS standard that defines a set of best practices for building and consuming RESTful APIs. OData helps you focus on your business logic while building RESTful APIs without having to worry about the various approaches to define request and response headers, status codes, HTTP methods, URL conventions, media types, payload formats, query options, etc.

OData RESTful APIs are easy to consume. The OData metadata, a machine-readable description of the data model of the APIs, enables the creation of powerful generic client proxies and tools.

See also

Working with OData in Lianja

Lianja	✔
Other	

39. Has world class full stack support for the complete platform

Don’t gamble your future and your customers future on products using languages and databases that are unsupported.

Lianja	✔
Other	

40. Provides a Service Level Agreement (SLA) that you can provide to your customers

The SLA should provide priority response for show-stopper and production issues and be tiered to guarantee response times 24x7 if required for business critical systems.

Lianja	✔