EIS Technical Solution Design Part 2

IV. LESSON PROPER

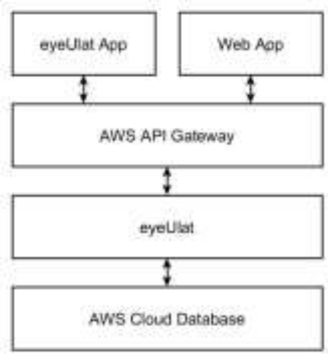
Solution Description

Defining the solution at the highest level of abstraction is akin to problem solving. I have used parts of the problem-solving model to get this high-level solution definition for years. It works and it is discussed here.

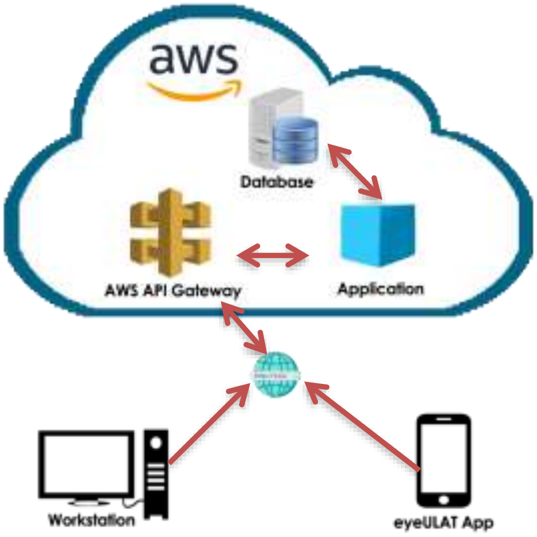
Example:

A. **Logical Architecture**

Architecture Level 2-3 diagram, showing affected/used components and interfaces



B. **High Level Architecture**



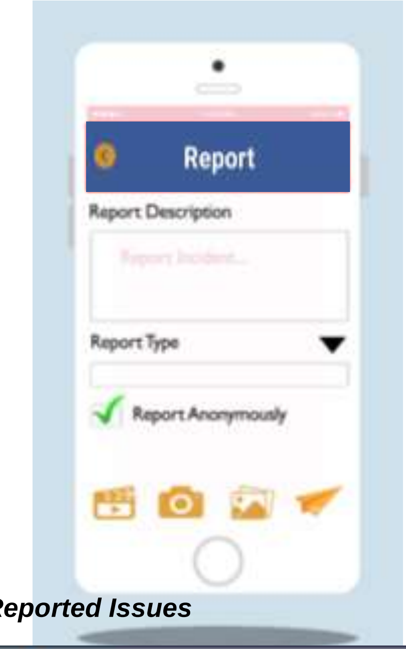
A representation of HL architecture used here to show how the systems will interconnect.

Process Flow

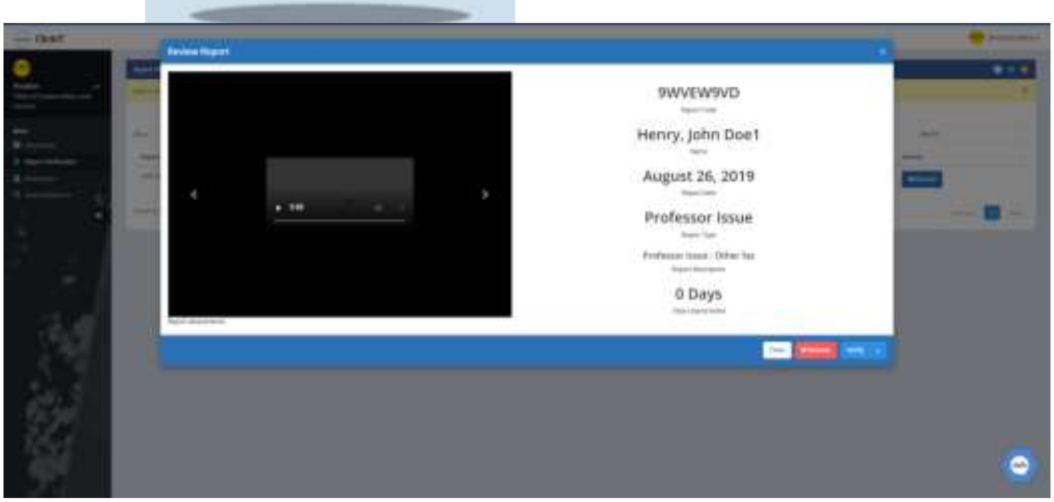
It is a method of visually documenting the stages involved in performing a certain business procedure. A type of process flow popular among business managers might take the form of flow charts that show inputs or information requests, followed by each of the steps required to create deliverable outputs such as products or services from the inputs.

Example:

Sending Reports



Monitor the Reported Issues



9WVEW9VD

Henry, John Doe1

August 26, 2019

Professor Issue

Professor Issue : Other Iss

0 Days

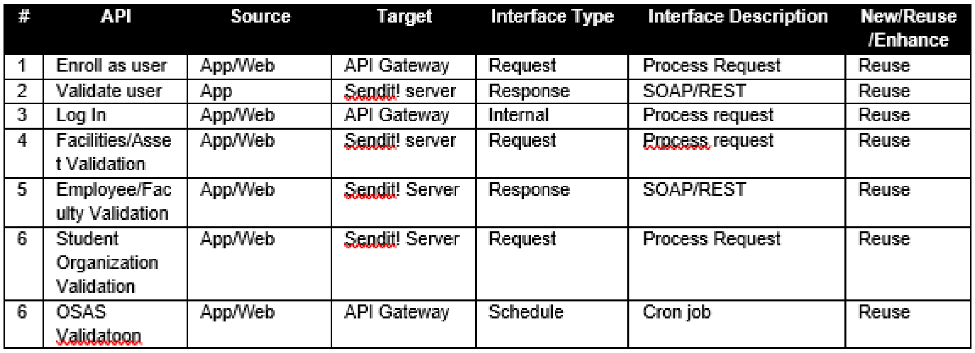
Redeeming Rewards

System Architecture

Solution Details

A **solution** is an implementation of people, processes, information and technologies in a distinct system to support a set of business or technical capabilities that solve one or more business problems.

Example:



#	API	Source	Target	Interface Type	Interface Description	New/Reuse /Enhance
1	Enroll as user	App/Web	API Gateway	Request	Process Request	Reuse
2	Validate user	App	Sendit! server	Response	SOAP/REST	Reuse
3	Log In	App/Web	API Gateway	Internal	Process request	Reuse
4	Facilities/Asset Validation	App/Web	Sendit! server	Request	Process request	Reuse
5	Employee/Faculty Validation	App/Web	Sendit! Server	Response	SOAP/REST	Reuse
6	Student Organization Validation	App/Web	Sendit! Server	Request	Process Request	Reuse
6	OSAS Validatoon	App/Web	API Gateway	Schedule	Cron job	Reuse

Security Assessment

Information Technology Security Assessment (IT Security Assessment) is an explicit study to locate IT security vulnerabilities and risks.

Purpose of Security Assessment

The goal of a security assessment (also known as a security audit, security review, or network assessment), is to ensure that necessary security controls are integrated into the design and implementation of a project. A properly completed security assessment should provide documentation outlining any security gaps between project designs and approved corporate security policies. Management can address security gaps in three ways: Management can decide to cancel the project, allocate the necessary resources to correct the security gaps, or accept the risk based on an informed risk / reward analysis.

Example:

> Solution provider must fully comply with Information Security requirements. Security controls should be implemented that insure the validation of input data, internal processing and output data.
>
> ****Low – resource utilization is 2hrs*

Testing Strategy

A **test strategy** is an outline that describes the testing approach of the software development cycle. The purpose of a test strategy is to provide a rational deduction from organizational, high-level objectives to actual test activities to meet those objectives from a quality assurance perspective. The creation and documentation of a test strategy should be done in a systematic way to ensure that all objectives are fully covered and understood by all stakeholders. It should also frequently be reviewed, challenged and updated as the organization and the product evolve over time. Furthermore, a test strategy should also aim to align different stakeholders of quality assurance in terms of terminology, test and integration levels, roles and responsibilities, traceability, planning of resources, etc.

Test strategies describe how the product risks of the stakeholders are mitigated at the test-level, which types of testing are to be performed, and which entry and exit criteria apply. They are created based on development design documents. System design documents are primarily used, and occasionally conceptual design documents may be referred to. Design documents describe the functionality of the software to be enabled in the upcoming release. For every stage of development design, a corresponding test strategy should be created to test the new feature sets.

Example:

> Testing was being done during each stage in the development. There is one (1) dedicated tester assigned on every sprint. The test cases for each sprint should be developed from the functional specification and use cases. Test Cases are based on the agreement during sprint planning of the Team depending on the timeline of a sprint. If a sprint will take 2 weeks, then it needs to determine how many can one (1) dedicated resource can execute in this timeline.