

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/29/2014 | 1.0 | Created | Jim Heary |
| 07/11/2014 | 2.0 | Revised | Jim Heary |
| 06/19/2015 | 3.0 | Revised for parking permits | Jim Heary |
|  |  |  |  |

Table of Contents

Introduction 4

Overview 4

Scope 4

Risks 5

Business Activities 5

Access Control 5

Time and Attendance 6

Create Badge 7

System Behavior 8

Overview 8

Actors 8

Use Cases 8

Create Employee 9

Update Employee 9

Create Vendor 9

Update Vendor 9

Link Template 9

Import Drivers 10

Business Rules 10

Non-functional Requirements 10

Presentation 10

Performance 10

Security 10

Availability 10

Concurrency 10

Interoperability 10

Key Abstractions 11

States 11

System Design 12

Overview 12

Design Model 12

Design Structure 12

Design Behavior 14

User Interfaces 14

Persistence 14

Availability 14

Concurrency 15

Security 15

Deployment 16

Appendix 18

Reports 18

# Introduction

## Overview

The Badge System enables the Argix Logistics HR department to create badges for Argix employees, drivers and vendors who need access to Argix Logistics facilities. The Badge System provides this information to a Lenel System that manages facility access control. In addition, the Badge System consumes information about drivers from the Argix Tsort System.

This document details known requirements and a candidate design. It does not address manpower requirements or project scheduling, although these can be determined from this architecture.

*How to read this document:*

Management- Introduction, Analysis (Overview, Business Activities, Actors), Project Planning

Operations- Introduction, Analysis, Project Planning

IT- Introduction, Analysis, Design, Project Planning

Context

The following diagram provides context for the system under discussion. This is a view of the system that helps us to see the system at a very high level. It helps us to understand the people (i.e. <<roles>>), hardware (i.e. <<devices>>), and external systems (i.e. <<systems>>) that interact with the system under discussion.



## Scope

Solution Boundary

The scope of the Badge System is limited to managing badge information for Employees and Vendors so that Badges can be created in the Lenel System. It is not concerned with Time and Attendance, which is handled by the ADP System, or Access Control, which is handled by the Lenel System. In addition, interoperability with the Tsort and Lenel Systems is being addressed outside of this architecture.

Constraints

1. The solution should conform to the principles of a Service Oriented Architectures (SOA).
2. The solution will use the current Microsoft-based platform and Microsoft-based development tools.

## Risks

None to address.

## Business Activities

The following diagrams show the business activities for the HR Systems. Business activities are modeled using activity diagrams. Activity diagrams show a series of activities, the actors responsible for each activity, and the information consumed and created during the process. These diagrams use the language of the business. Activity diagrams drive the discovery of Actors, Use Cases, and Key Abstractions.

Access Control

The following activity diagram shows the activities involved with control of access to an Argix Logistics facility. The Badge Holder (i.e. Employee, Driver, or Vendor) presents his Badge to a Badge Scanner to gain access to a facility (I.e. terminal). The Badge Scanner reads the Badge and presents the information to the Lenel System to validate facility access. The Lenel System validates the Badge Holder and returns access granted or denied to the Badge Scanner. The Badge Scanner unlocks the door if access has been granted; otherwise, access is denied. If an error occurs while scanning the Badge, then access is denied.

This activity diagram is intended to provide additional context for the Badge System; but, it is outside the scope of the system architecture.



Time and Attendance

The following activity diagram shows the activities involved with recording time and attendance for an Argix Employee. An Employee presents her hand to a Bio-Scanner upon beginning and ending her shift. The Bio-Scanner reads biometric information and determines the Employee. If the Employee is found, attendance information is presented to the ADP System. The ADP System creates (upon arrival) or updates (upon departure) a record of attendance for the Employee.

This activity diagram is intended to provide additional context for the Badge System; but, it is outside the scope of the system architecture.



Create Badge

The following activity diagram shows the activities involved with creating a Badge for a new Employee (could also be a Vendor). The HR Assistant enters the badge information into the Badge System. The Badge System creates a new Employee Badge that is in an active state. At a regularly scheduled time, the Lenel System pulls the new Badge information from the Badge System and saves the Badge information. When the Badge appears in the Lenel System the HR Assistant can choose to create the physical Badge.

*Notice the Badge System does nothing more than provide CRUD-like behavior for managing Badge information.*

**NOTE**: Badge information from the Badge System contains a department field. This field is linked in the Lenel System to a Badge Template which specifies the look and information that is created on the physical Badge. If a new department is created in the Badge System, then the Lenel System will associate the new department with the *default* Badge Template. The IT department is responsible for managing this association.



# System Behavior

## Overview

The purpose of defining system behavior is to discover, capture, and analyze the requirements of the system under discussion. This is achieved by describing the requirements (i.e. the conditions or capabilities to which the system must conform) well enough so that an agreement can be reached between the business users and the system developers on what the system should and should not do. It begins by modeling the business processes with a series of activity diagrams. These diagrams drive discovery of the users of the system (i.e. Actors), the system functionality (i.e. Use Cases), and the vocabulary of the system (i.e. Key Abstractions). From these artifacts, an analysis model is created that drives system design and development.

## Actors

An actor specifies a role played by a user or any other system that interacts with the system under discussion. Actors influence UI design and security concerns. The list below provides definitions for the actors surrounding the Badge System.

* Badge System- the system under discussion; stores information needed to create physical Badges.
* IT Admin- a member of the IT group that associates the Lenel System Badge Templates with the Badge data department fields.
* HR Assistant/HR Manager- employee of Argix Logistics that manages Badge information and creates physical Badges for badge holders.
* Tsort System- internal Argix Logistics system that is a source of Driver badge information.
* Lenel System- third-party software system used by Argix Logistics to manage facility access and for creating Badges.
* ADP System- third-party software system used by Argix Logistics to manage time and attendance.
* Badge- a physical badge worn by Employees, Drivers, and Vendors.
* Badge Maker- a device for creating physical Badges.
* Badge Scanner- a device for scanning physical Badges for facility access. It is connected to the Lenel System.
* Bio-Scanner- a device for scanning Employee hand for time and attendance. It is connected to the ADP System.
* Driver- a driver for Argix Logistics; wears a Badge for facility access.
* Vendor- a vendor of Argix Logistics; wears a Badge for facility access.
* Employee- an employee of Argix Logistics; wears a Badge for facility access.

## Use Cases

A Use Case is a list of steps, typically defining interactions between a role (i.e. Actor) and a system to achieve a goal. The actor can be a human or an external system. Use Cases describe the functional view of the system under discussion as a set of business transactions. Use Cases influence UI design, domain models, application service interfaces, and define business transactions. The following Use Case diagram shows some, if not all, of the actors and use cases involved in Badge System.

The following use case diagram shows the use cases that are scoped to the Badge System. These use cases are derived from the activity diagram Create Badge.



Create Employee

The HR Assistant needs to create a new Employee. The user enters Employee information including last name, first name, middle name, suffix, social security number, hire date, location, sub-location, department, and status.

If the Employee is validated, then the system creates a new Employee (active).

If a new Employee department is needed, then proceed to use case Link Templates.

Update Employee

The HR Assistant needs to update an existing Employee. The user updates Employee information including last name, first name, middle name, suffix, social security number, hire date, location, sub-location, department, and status.

If the Employee is validated, then the system updates the Employee.

Create Vendor

The HR Assistant needs to create a new Vendor. The user enters Vendor information including last name, first name, middle name, suffix, location, department, and status.

If the Vendor is validated, then the system creates a new Vendor (active).

If a new Vendor department is needed, then proceed to use case Link Templates.

Update Vendor

The HR Assistant needs to update an existing Vendor. The user updates Vendor information including last name, first name, middle name, suffix, location, department, and status.

If the Vendor is validated, then the system updates the Vendor.

Link Template

The IT Administrator needs to link a Lenel System Badge Template to a department used by Employees or Vendors. This is done in the Lenel System after a new department has been created in the Badge System and then replicated to the Lenel System.

Import Drivers

The Badge System needs to import new Drivers from the Tsort System into the Badge System. This use case is addressed outside of this architecture.

## Business Rules

1. The Badge::Department list cannot be changed without the IT department.
2. Employee and Vendor Badge departments must be unique.

## Non-functional Requirements

Presentation

Thin client user interface accessible from desktop only.

Performance

Unspecified.

Security

Authentication

Employees will be authenticated by the Argix network domain.

Authorization

Employees will be authorized per role-based membership (i.e. HR Assistant) using domain accounts.

Availability

Business hours.

Concurrency

Support 1-2 simultaneous users.

Interoperability

Lenel- one-way communication initiated and controlled by the Lenel System.

Tsort- one-way communication initiated and controlled by the Tsort System.

## Key Abstractions

The class diagram below shows the key abstractions involved in Badge System. Key abstractions are the key concepts and abstractions that the system needs to handle. They are those things that, without which, you could not describe the system. Key abstractions drive design of the database schema and the domain model (if applicable).



* BadgeData- base information for Badge data in the Badge System.
* DriverBadgeData- driver-specific information for Driver badge data; inherits from BadgeData.
* EmployeeBadgeData- employee-specific information for Employee badge data; inherits from BadgeData.
* VendorBadgeData- vendor-specific information for Vendor badge data; inherits from BadgeData.

## States

The state machine view describes the dynamic behavior of objects over time by modeling the lifecycles of objects of each class. Each object is treated as an isolated entity that communicates with the rest of the world by detecting events and responding to them. Events represent the kinds of changes that an object can detect. Anything that can affect an object can be characterized as an event.

There are no states to discuss in the Badge System.

# System Design

## Overview

The system design requires a thin web client for Internet access by Windows domain users. The web client will access back-end databases through web services arranged in a Service Oriented Architecture (SOA). Integration with other systems is done at the database level and outside of the scope of this architecture. Security will be provided by an ASP.Net role database that hosts Argix domain accounts. Application services, both user interface and web service, will be secured using pre-defined user roles (i.e. HR Assistant).

## Design Model

Design Structure

The design is an n-tier physical and logical design composed of an ASP.NET 4.0 Web Forms client (Argix10.IDBadges.Web), a .NET 4.0 WCF Services (Argix10.HR.Services) web service, and a SQL Server 2012 R2 database. The design as a whole is built to conform to a Service Oriented Architecture. Service Oriented Architectures describe ways to build loosely-coupled systems composed from individual services. Application integration is accomplished at the database level using batch SQL Jobs. All software components are built with Microsoft Visual Studio 2010 and compiled to operate in.Net 4.0 framework environments (i.e. servers and desktops).

Enterprise Patterns

In [software engineering](http://en.wikipedia.org/wiki/Software_engineering), a [design pattern](http://en.wikipedia.org/wiki/Design_pattern) is a general reusable solution to a commonly occurring problem within a given context in [software design](http://en.wikipedia.org/wiki/Software_design). It is a description or template for how to solve a problem that can be used in many different situations. Patterns are formalized [best practices](http://en.wikipedia.org/wiki/Best_practice) that the programmer must implement in the application. [Object-oriented](http://en.wikipedia.org/wiki/Object-oriented) design patterns typically show relationships and [interactions](http://en.wikipedia.org/wiki/Interaction) between [classes](http://en.wikipedia.org/wiki/Class_(computer_science)) or [objects](http://en.wikipedia.org/wiki/Object_(computer_science)), without specifying the final application classes or objects that are involved. The following enterprise design patterns are used throughout the design:

Transaction Script- a Transaction Script organizes business logic by procedures where each procedure handles a single request from the presentation. They are an excellent choice when we don’t have a middle tier domain model. Transaction scripts are used in the web services to provide business transactions and security for one or more use cases.

Gateway- a Gateway is an object that encapsulates access to an external resource such as a web service or database server. Gateways are used throughout the design for user interface access to middle tier web services, for web service access to backend database services, and for access to external hardware.

Record Set- a Record Set is an in-memory representation of tabular data; record sets (i.e. DataSet) work well with .NET user interface components.

Data Transfer Object- a Data Transfer Object (DTO) is an object that carries data between processes in order to reduce the number of method calls. Data Transfer Objects are serialized classes exposed by the interfaces of the web services for transactional calls to the middle tier.

Components and Interfaces

The following drawing shows the components and interfaces of the Badge System. This is a logical view of the main components, classes, and interfaces that make-up the structure of the design. It does not attempt to show all components, classes, and interfaces that make-up the complete implementation, only those that are architecturally significant.



Argix10.IDBadges.Web

The user interface component, Argix10.IDBadges.Web, is an ASP.NET Web Forms application that uses a master-detail paradigm. The Badges web form uses the asp:GridView grid control to display Employee and Vendor Badge data. A new badge can be created, or an existing one updated, using the Badge web form, which is a simple HTML form. The forms communicate with a WCF Service, Argix10.HR.Services, through the HRGateway gateway. Recordset BadgeDataset provides typed information to the web forms for data binding.

Argix10.HR.Services

The web service component, Argix10.HR.Services, exposes the BadgeService service, a transaction script that implements the IBadgeService interface. A transaction script is adequate in this design since a domain model is not required; reference the key abstractions to see that there is only one notable abstraction, BadgeData. The IBadgeService exposes CRUD-like methods for creating and updating Employee and Vendor badge data. The transaction script also supports transactions (although none are required in this case) and method-level security. Data Transfer Objects (i.e. <<dto>>) are used to upload badge data in create and update operations. The BadgeService is secured using role based service behaviors; this requires the wsHttpBinding with Message security and client credential type Windows.

ArgixEmployee/ArgixVendor Database

Persistence is maintained in two backend SQL Server databases. Each database persists to a single table, BadgeData, exposing services with SQL stored procedures.

The design does not attempt to show how interoperability is achieved with the Tsort System or the Lenel System as these are outside the scope of this architecture.

Design Behavior

There is no notable behavior to discuss since the application offers simple CRUD services and there are no workflows associated with the business processes.

User Interfaces

Web Client

The user interface exposes a simple master-detail web front end. The web forms are coded in HTML5, use CSS3 for positioning and styling, and use Ajax to provide a richer user experience. Ajax (short for asynchronous [JavaScript](http://en.wikipedia.org/wiki/JavaScript) + [XML](http://en.wikipedia.org/wiki/XML)) is a group of interrelated [Web development](http://en.wikipedia.org/wiki/Web_development) techniques used on the [client-side](http://en.wikipedia.org/wiki/Client-side) to create [asynchronous](http://en.wikipedia.org/wiki/Asynchronous_I/O) [Web applications](http://en.wikipedia.org/wiki/Web_application). With Ajax, Web applications can send data to, and retrieve data from, a [server](http://en.wikipedia.org/wiki/Web_server) asynchronously without interfering with the display and behavior of the existing page. In addition, jQuery is used for client-side behaviors such as input masks for date fields. jQuery is a [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) [JavaScript library](http://en.wikipedia.org/wiki/JavaScript_library) designed to simplify the [client-side scripting](http://en.wikipedia.org/wiki/Client-side_scripting) of [HTML](http://en.wikipedia.org/wiki/HTML), and is [free, open source software](http://en.wikipedia.org/wiki/Free_and_open_source_software) licensed under the [MIT License](http://en.wikipedia.org/wiki/MIT_License). Role-based security authorizes the HR Assistant/HR Manager to use services for creating and updating Employee and Vendor Badge data, and authorizes IT Administrators to create new departments.

Persistence

Database Schema

The database schema is composed of a single table of badge data for each Badge type (i.e. Employee, Vendor, and Driver). A few other tables exist to support lists for locations, sub-locations, departments, and status. The AccessControl database is the single point for information exchange between the Lenel System and the BadgeData databases. Information is moved between these tables and the AccessControl table with SQL jobs triggered by time-based schedules.



## Availability

As provided by the platform.

## Concurrency

None required.

## Security

Authentication

Users are authenticated by the network upon login to their Windows machine.

Authorization

The web application uses ASP.NET role-based security to authorize access to application features and services. Each web form has access to global objects accessible from the System.Web.Security .NET component. These global objects are made accessible to the application by adding security entries to the ASP.Net configuration file (i.e. web.config) including <authentication> and <roleManager> sections of <system.web>.



## Deployment

The current deployment consists of a centralized application on the Jamesburg local area network (LAN) with client computers running on the Jamesburg LAN as well. Access to the application will be via the Argix Corporate Portal (i.e. SharePoint) by exposing a link to the web application. The following drawing shows the application components deployed to a Microsoft-based platform.



Desktop- the desktop requires a web browser, such as Internet Explorer 10+.

Web Server- IIS 7.5 is required to host the IDBadges ASP.Net web application and the Argix10.HR.Services WCF service. Deployment is as follows:

IDBadges

1. Deploy to c:\inetpub\wwwroot\Argix10\IDBadges windows folder
2. Convert to an IIS Application with an ASP.NET v4.0 application pool with the Classic managed pipeline mode and the ApplicationPoolIdentity identity
3. Authentication: disable Anonymous Authentication, enable ASP.Net impersonation, and enable Windows Authentication

Argix10.HR.Services

1. Deploy to c:\ inetpub \wwwroot\Argix10\Argix10.HR.Services windows folder
2. Convert to an IIS Application with an ASP.NET v4.0 application pool with the Integrated managed pipeline mode and the ApplicationPoolIdentity identity
3. Authentication: enable Anonymous Authentication (default)

# Appendix

## Reports

None