ATLANTIX GLOBAL Web Portal

7/15/2014

Contents

Glossary

- **Admin** – Refers to a user with editing capabilities. Admin will manage all Users and Companies information.
- **Atlantix Database** – Atlantix's DB2 database for managing the business data. It is a database which is responsible for the majority of data management, as described in the data table diagram below. Each of these tables will have a Stored Procedures (SP), which may be called to retrieve or post data. Connection occurs via Java connect to DB2 and provided by Atlantix.
- **Atlantix platform** – Atlantix web portal pulls data for local storage from 1 location. Exchange between DB2 and Web portal will be done by a Stored Procedures (SP).
- **Create New Job** – Create New Job handles all jobs creation for pickups and will include following information: Location details, Job Contracts, Equipment details, documents, report details.
- **Dashboard** – allows users to view latest activity from all files by consolidating and display data via the tables cluster
- **Database/DB2** - DB2 (AS400) database version V6R1
- **Login Screen** – allows users to login to the web portal by providing a user name and a password; and contact for the support
- **Pickups –** Pickups handles all pickups information (ie. Job type, company name, location, dates , status, etc)
- **Prepared Reports –** Prepared Reports handles all stored documents, which can be exported and printed.
- **Reporting Tool –** Reporting Tools handles inventories with Atlantix, Inventory Sold, Open Jobs
- **Job ID:** Also referred to as PTN, Report Request ID, Job #,
- **SP/Stored Procedure:** Store Procedure used to pull and push data between web portal and DB2.
  - **GET:** Specifies object/data to retrieve (i.e. By ID, by email) from DB2 via SP.
  - **Local:** Specifies which fields will be stored locally (mysql).
  - **POST:** Specifies fields to be updated in DB2 via specified request and corresponding SP[1].
    - [1]*Updates/Edits of existing data in a single field.*
  - **PUT:** Specifies **required** field to create an object, for form validation and SP[2].
    - [2]*Creates object in DB2 with corresponding required fields.*
  - **Request:** Is the name of the SP
  - **(x):** Means the field is optional
  - **x:** Means the field is required
- **Stored Documents –** Stored Documents handles all stored documents, which can be exported and printed. Locally stored files and data.

Company Description

Based out of Atlanta, Georgia, ATLANTIX Global as a business is one of the world's largest brokers of used IT equipment in the world with revenues between 100mm and 130mm annually. Considerable amounts of secondary and used hardware are sold globally by ATLANTIX, with large influxes of business coming from major technology companies (Google, Microsoft, Yahoo, Facebook, etc) who are offloading large amounts of 3 to 5 year old gear. For example, Google may send 40,000 servers which ATLANTIX will then process and sell over the next 6 to 12 months.

ATLANTIX Global has one primary warehouse and a multitude of warehousing partners and is also a Dell premier partner reselling Dell and other manufacturer's products. ATLANTIX has a constant flow of business and an ever-changing inventory. Massive quantities of pre-owned gear are held at any given time (over 8mm inventory with a list price well north of 50mm) with most parts being sold out domestically and other inventory being broken down into components and then sold out internationally.

ATLANTIX provides its customer with the ability to schedule pickups, which is useful for larger orders, and also provides its customers with access to on-demand reporting. The market that ATLANTIX occupies is one of significant opportunity and growth as technology continues to advance rendering older hardware obsolete, especially in cutting edge tech companies.

Technical Requirements

- **OS:** Windows 2008 R2
- **HTTP:** Apache2
- **Backend Language:** PHP 5.4 (5.4.7)
- **Database:** MySQL 5.0 (5.0.1)
- **Frontend Languages:** HTML5, CSS3, Javascript
- **Frontend Libraries:** Google Fonts API, JQuery 2 (or latest compatible version), HTML5Shiv
    - **Framework:** Laravel 4.2
        - Full Release notes can be found here: http://laravel.com/docs/releases#laravel-4.2
        - Framework includes use of the following:
        - Mailgun and Mandrill API drivers for mail functions
        - Artisan CLI
        - Composer (Latest compatible version, unlisted)
        - Bootstrap 3 (3.1.1)
        - “Homestead” and “Cashier” do not apply for this project.
- **Recommended Components:** Performance components for load time (ie. caching) such as Gzip and Memcached are recommended but not required.
- **Required Components:**
    - PHP IBM DB2 Functions (PECL extension). Setup, requirements, and usage documentation found here: http://www.php.net/manual/en/install.windows.php (http://www.php.net/manual/en/install.windows.apache2.php)
        - Required IBM DB2 Universal Database client installed on the PHP/MySQL web application server.
        - This doesn’t mean the DB2 DB needs to be hosted locally, a remote DB host is acceptable.
        - Use of Use of DB2\400 version V6R1 required and confirmed technically compatible by client.
        - Please review provided documentation for installation procedures and requirements.
- **Firewall:** FTP and RDP accessible over any preferred port and method. FTP should be restricted to the site’s root directory, if possible, for easy of post-launch updates. Static IP’s can be provided for restriction, if needed (65.112.118.122). Dynamic IP’s can be provided to sys admin on an as-needed basis or VPN access may be permitted. SMTP port should be opened for any mail functions that may exist in scope (if any).
- **DB2 Integration:** DB2 Stored Procedures to be executed via PHP’s IBM DB2 Functions (see required components). Technical specification document to be provided at end of planning will detail exchange of data.

User Flows

Portal has 3 main user flows, with alternative layout for customer user:

1. Atlantix Admin
2. Customer Admin
3. Customer User (created by Customer Admin)
    - 3 frontend templates to be assigned, in admin panel, by user type 1 and/or 2.

Integration

1. BFM creates a local MySQL database for interaction with the web portal
    - Web portal pulls data for local storage from 1 locations:
        - DB2
            - Connection occurs via Stored Procedures within DB2, to be provided by Atlantix.
            - DB2 (AS400) version V6R1 database
            - Local storage used for faster queries
    - Stored Procedures used to pull and push data between web portal and DB2. SP's to be provided from Atlantix, to BFM to execute get and post data.
2. Web Portal receives confirmation from DB2 of data exchange.
3. Interaction between ecommerce and DB2 is not in scope (this currently exists).

Frontend Layouts and corresponding backend modules:

Please note that the "GET" and "PUT" considerations are recommended for clarity; a details specification of data exchange is documented in the DB2 Integration section, below.

1. Login / Logout functionality (assumed no registration, all users to be provided by Atlantix via integration: GET)
2. Dashboard (report data via integration: GET)
3. View Pickups (table layout with search, sort, filtering; data via integration: GET & PUT)
    a. Ability to edit (on point 4 layout), delete, and re-submit existing
4. Create New Job (pulls available data for dropdowns, posts data via integration: GET & PUT)
    - Location Details Form (tab)
    - Job Contacts Form (tab)
    - Equipment Details Form (tab)

- Scheduling Details Form (tab)
- Documents? (tab)
  - No specs on this provided, not considered in scope.
    - Custom Spreadsheets and Word Docs
- Report Details Form (tab)

5. Reporting Tool
   - Has tabs which replicate from Pickup and Job modules above (3 and point 4 above)
   - Reports tab
     - Table layout (pulls data from reporting integration: GET)
6. Prepared Reports Tool (and Stored Documents)
   - Table layout with XLS downloads (GET)
     - ASSUMPTIONS:
       - It is assumed that these filed are already “prepared” and we are only providing ability to download these.
       - It is assumed that these are provided in a directory for us to be able to prepare data (sort data, size, etc.) for. Otherwise it is assumed this data is provided to us (ie. via SP, XML, log, etc.)
       - It is assumed that the two tools are the same (and that the only difference is that “selected and exported” docs are saved to the user, for later view, via “Stored Documents” tab.
7. Admin
   - Manage Companies Tab (GET)
     - Table layout, pulls data for display purposes
   - Manage Users Tab (GET & PUT)
     - Ability to create new user and edit existing users.
       - Form contains basic user data for editing (location, subscription, etc.)
       - Form contains ability to assign a specific layout to the corresponding user.

Target Consumer/Client:

- Sellers of gently used and pre-owned technology hardware and components
- Buyers of gently used and pre-owned technology hardware and components
- Enterprise level technology companies (Google, Microsoft, Yahoo, Facebook, etc)

Intended User Behavior

- Access status updates on their gear (pickup, purchase price, sale price)
- Schedule new pickups and make new requests
- Obtain information needed for compliance

Key Features:

- Integration with DB2 AS/400 Database via Stored Procedures
- Company dashboard
- Customer ability to view jobs/pickups
- Customer ability to create/edit new jobs
- Customer ability to view/generate reports
- Customer ability to download prepared reports
- Customer ability to download stored documents
- Admin ability to add/edit/manage companies
- Admin ability to add/edit/manage users

UI/UX Considerations

User Creation

- No passwords will be stored locally. Logic for forgot password is listed below.
- Create new user form will trigger email, upon submission, to user to notify them of new account, via application. PHP mail function will retrieve password via form GET from text field in “create new user” form, without local storage.

Forgot Password?

In the case that a password needs to be retrieved this is the recommended application flow:

(1) User enters his/her email address
(2) A token is generated and send to user email address
(3) User goes to the token page by clicking on link provided in the email
(4) A form to create a new password is presented
(5) Application communicates with SP to update password
(6) After successful password reset, the user is prompt to login to the application
(7) An email is sent to the user notifying of the password reset.

This ensures that no password is transmitted or stored. It’s just updated in the SP

Responses

The expected response should include a Success flag, (0 or 1), an Error Message in case of an unsuccessful request and a collection of objects, For example: SP_User GET request should return a collection of 1 or more Users with all the fields specified in the SP_User fields.

If for GET requests, no parameter is specified, it should return a collection of objects as well as a page flag and a limit for pagination purposes. Parameters page and limit can be included as part of the request parameters to limit the results.

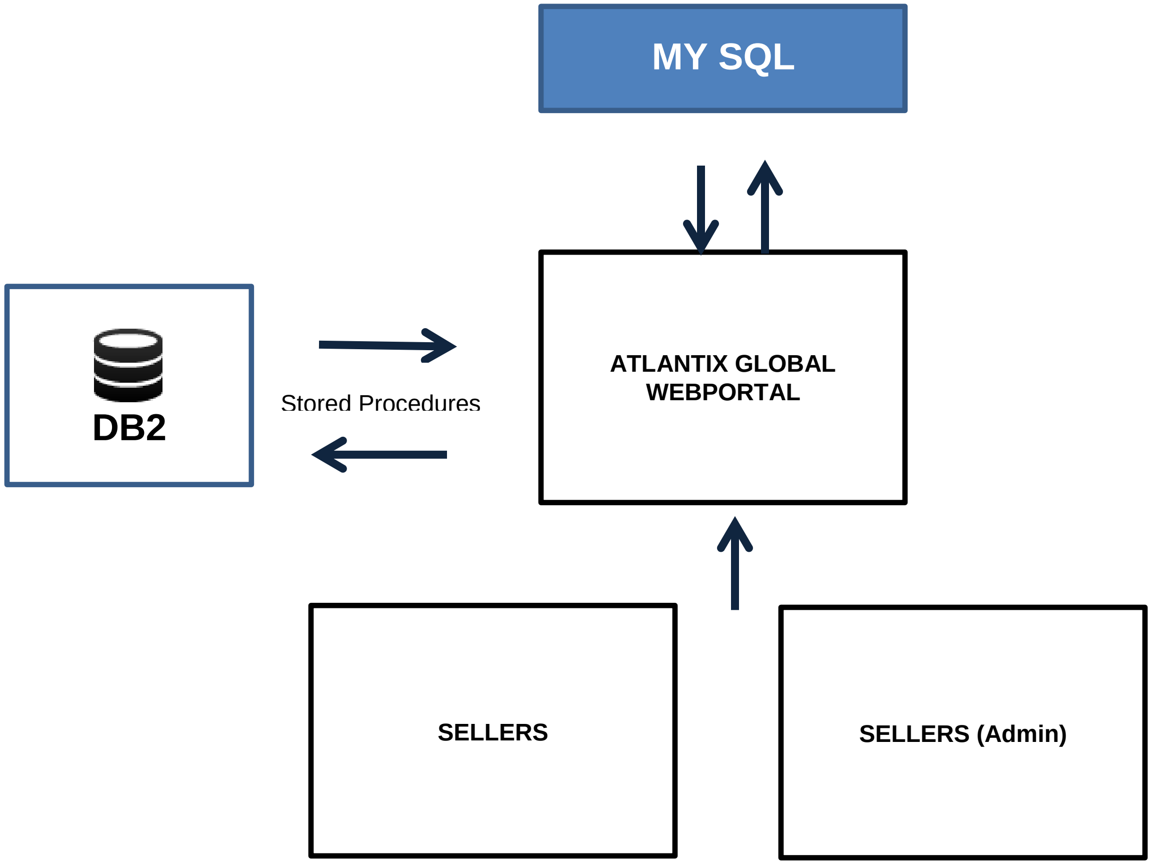
All objects should include a created and updated field.

Further Considerations

- Remove “delete” and “refresh” buttons (currently present in client provided mockups). All refreshing of data will occur on page load for corresponding module.
- Report documents to be uploaded via Atlantix admin file repository (ie. “media library”) with ability to assign to available user accounts and companies. Files to be locally stored, only.
- All “Export to XLS” functions should export to XLSX.
- Atlantix admin persona will utilize same interface as customer admin persona with exception to greater editing capabilities and ability to select from all available companies and users.
- Upon create new job submission, do not store locally, simply trigger DB2 SP to get the Report Request ID from Atlantix. Should display in Report Details view upon completed submission.
- Local storage for search to be triggered by SP upon view jobs page load (SP required for retrieval of all jobs).
- Primary Key will be local AI for application, rather than the “Report Request ID” alone.

DB2 Integration

Diagram



Tables, Fields and Methods Table

Request: SP_Login

Field	Type	Local	GET	POST	PUT	Comments
Email	String	x		x		
Password	String			x		

Request: SP_Users

Field	Type	Local	GET	POST	PUT	Comments
ID	Integer	x	(x)			Should be generated by SP upon PUT Request
First Name	String	x	(x)	(x)	x	
Last Name	String	x	(x)	(x)	x	
Company	Company	x	(x)	(x)	x	See SP_Companies for fields specification
Email	String	x		(x)	x	
Password	String			(x)		Should be generated by SP upon PUT Request This should be returned only on PUT requests
Phone	String	x		(x)	x	Formatted phone (###) ###-####
Mobile	String	x		(x)	(x)	Formatted phone (###) ###-####
Fax	String	x		(x)	(x)	Formatted phone (###) ###-####
Location	Location	x	(x)	(x)	(x)	See SP_Locations for field specification
Status	Boolean	x		(x)	x	

Role	Integer	x	(x)	(x)	x	1: Admin 2: Seller
All Locations	Boolean	x		(x)	x	
JobCreated	Boolean	x		(x)	x	Notification flag
JobRequested	Boolean	x		(x)	x	Notification flag
JobScheduled	Boolean	x		(x)	x	Notification flag
JobCompleted	Boolean	x		(x)	x	Notification flag
JobReport Sent	Boolean	x		(x)	x	Notification flag
JobCancelled	Boolean	x		(x)	x	Notification flag

Request: SP_Jobs

Field	Type	Local	GET	POST	PUT	Comments
ID	Integer	x	(x)			Should be generated by SP upon PUT Request
Job Number	Integer	x	(x)			Job number or PTN Should be generated by SP upon PUT Request
BOL	Integer	x				Should be generated by SP upon PUT Request
Pickup Date	Date	x	(x)	(x)	x	i.e. 2014-07-10 13:02:01 -0000 GET should allow a range
Status	Integer	x	(x)	(x)	x	1: Created 2: Requested 3: Scheduled 4: Landed

						5: Recorded 6: Processing 7: Completed 8: Cancelled
Client Ref Number	String	x	(x)			i.e. AK-0001 Should be generated by SP upon PUT Request
*** Dock Date**	Date	x		(x)		i.e. 2014-07-10 13:02:01 -0000 Should be added after job creation / Status update
*** Freight Career**	String	x		(x)		Should be added after job creation / Status update
*** Receipt Date**	Date	x		(x)		i.e. 2014-07-10 13:02:01 -0000 Should be added after job creation / Status update
*** Pieces**	Integer	x		(x)		Should be added after job creation / Status update
*** Requested Weight**	Decimal	x		(x)	x	
*** Description**	String	x			x	
Origin	Location	x	(x)	(x)	x	See SP_Locations for fields specification
*** Destination**	Location	x	(x)	(x)	x	See SP_Locations for fields specification
*** Airway Bill Number**	String	x	(x)			Should be generated by SP upon PUT Request
*** Smart Way**	Boolean	x				0 or 1
Requested Date	Date	x	(x)		x	i.e. 2014-07-10 13:02:01 -0000 GET: Should allow a range
Type	String	x	(x)		x	Delivery / Pickup
Company	Company	x	(x)		x	See SP_Companies for fields specification

*** Landed Weight**	Decimal	x	(x)		
*** Net Amount**	Decimal	x	(x)		
Requested By	String	x		x	Full Name
Contacts	Contacts Collection	x	(x)	x	See SP_Contacts for fields specification
Equipment	Equipment	x	(x)	x	See SP_Equipment for fields specification
Earliest Date	Date	x	(x)	x	i.e. 2014-07-10 13:02:01 -0000
Latest Date	Date	x	(x)	x	i.e. 2014-07-10 13:02:01 -0000
Time of Day	String	x	(x)	x	
Notes	String	x	(x)	x	
Prepared Reports	PreparedReports Collection	x			See SP_PreparedReports Specification Should be added in SP_PreparedReports
Serial Number	String	x			Should be generated by SP upon PUT Request
Tracking Number	String	x	(x)		Should be added after job creation / Status update
Amount Due To Customer	Decimal	x	(x)		Should be added after job creation / Status update
Parts	Parts Collection	x	(x)	x	See SP_Parts for field specification
Parts Description	String	x	(x)	x	
Special Request	String	x	(x)	x	
Parts Document	Stored Document	x	(x)	x	See SP_Documents for field specification

Request: SP_Contacts

Field	Type	Local	GET	POST	PUT	Comments
ID	Integer	x	(x)			Should be generated by SP upon PUT Request
Name	String	x		(x)	x	
Email	String	x		(x)	x	
Phone	String	x		(x)	x	Formatted phone (###) ###-####
Type	String	x		(x)	x	Primary Pickup Alternate
Subscribed	Boolean	x		(x)	x	0 or 1
Job	Job	x	(x)		x	See SP_Jobs for field specification
Company	Company	x	(x)		x	See SP_Companies for field specification

Request: SP_Parts

Field	Type	Local	GET	POST	PUT	Comments
ID	Integer	x	(x)			Should be generated by SP upon PUT Request
Manufacturer	String	x		(x)		Should be added after job creation / Status update
Part	String	x		(x)	x	
Serial	Integer	x		(x)		Should be added after job creation / Status update
Job	Job	x	(x)		x	
Sold	Integer	x	(x)	(x)		Should be added after job creation / Status update
Quantity	Integer	x		(x)	x	

Request: SP_PreparedReports

Field	Type	Local	GET	POST	PUT	Comments
ID	Integer	x	(x)			Should be generated by SP upon PUT Request
URL	String	x			x	URL to download file Should be generated by SP upon PUT Request
Name	String	x			x	
Created	Date	x			x	i.e. 2014-07-10 13:02:01 -0000
Type	String	x			x	
Size	Integer	x			x	Bytes
Job	Job	x	(x)		x	See SP_Jobs for field Specification

Request: SP_Documents

Field	Type	Local	GET	POST	PUT	Comments
ID	Integer	x	(x)			Should be generated by SP upon PUT Request
URL	URL	x			x	Download URL Should be generated by SP upon PUT Request
Name	String	x			x	
Created	Date	x			x	i.e. 2014-07-10 13:02:01 -0000
Type	String	x			x	
Size	Integer	x			x	Bytes
Job	Job	x	(x)		x	See SP_Jobs for field Specification

Request: SP_Companies

Field	Type	Local	GET	POST	PUT	Comments
ID	Integer	x	(x)			Should be generated by SP upon PUT Request
Name	String	x		(x)	x	
Active	Boolean	x	(x)	(x)	x	0 or 1
Website	URL	x		(x)	x	
Location	Location	x		(x)	x	See SP_Locations for field Specification
Account Exec	User	x		(x)	x	See SP_Users for field Specification
Account Manager	User	x		(x)	x	See SP_ Users for field Specification
Contact	Contact	x		(x)	x	See SP_Contacts for field Specification
Billing Contact	Contact	x		(x)	x	See SP_ Contacts for field Specification

Request: SP_Locations

Field	Type	Local	GET	POST	PUT	Comments
ID	Integer	x	(x)			Should be generated by SP upon PUT Request
Address 1	String	x		(x)	x	
Address 2	String	x		(x)	x	
City	String	x		(x)	x	
State	String	x		(x)	x	
Country	String	x		(x)	x	

Postal Code	String	x		(x)	x	
Phone	String	x		(x)	x	Formatted phone (###) ###-####
Status	Boolean	x		(x)	x	
Company	Company	x	(x)		x	See SP_Companies for field specification
Contact	Contact	x		(x)	x	See SP_Contacts for field specification