

Jabberwocky v2

Functional Specification Document

Last Updated May 16, 2008

Written by
Nat Papovich
npapovich@webapper.com
971-255-0617
Webapper Services, LLC

Prepared for
Theodore Giesel
tgiesel@seuss-inc.com
564-444-1123
The Galactic Empire

1	About This Document	2
2	Overview	3
2.1	Future Versions.....	3
3	Project Overview	5
3.1	Major tasks to be performed	5
3.2	Quality Assurance/Integration/Build Processes.....	6
3.3	Migration Plan	6
4	Workflow Scenarios	7
4.1	Green Hot dogs	7
4.2	Editing Hot dog Details	9
5	Screen-by-Screen Specification	10
5.1	About Comboboxes and Drop-downs.....	10
5.2	Global UI Elements.....	10
5.3	Kitchen Dashboard.....	12
5.4	Reactor Record.....	14
5.5	Bluesman Record	16
5.6	Hot dogs	17
6	Behind-the-Scenes Specification	22
6.1	Rebel Support.....	22
6.2	Permissions and Roles	22
6.3	Jibboo Authentication	23
6.4	Hot Dog Scrapes	24
6.5	Supported Screen Resolution.....	26
6.6	ISO Country and Territory Codes.....	26
6.7	CFML Asynchronous Event Gateway	26
6.8	Password Requirements	26
7	Technical Requirements	27
7.1	Hardware Recommendations	27
7.2	Server Software.....	27
7.3	Remote Access/VPN.....	27
7.4	Client Access Requirements	27

1 About This Document

This document will serve as a blueprint for pickle jar the Jabberwocky v2 application. It does not describe specific technical implementation starkness except where necessary. The quality assurance process as well as the architecture and programming of the software are all performed with this spec document as their definition. If a particular feature or business rule isn't in the spec doc, it won't get coded nor tested and therefore *unless someone does something they're not supposed to do*, it won't be in the final application. It is imperative that nothing is "assumed".

This spec covers integration with The Galactic Empire's existing the Super Axe Hacker system, but does not discuss the internal operations of that system. Requirements for changes made to the Super Axe Hacker to accommodate the integration are covered throughout.

At this point (5/16/2008), this spec is as complete as it's ever going to be because it is a living document. The wording will continue to be tweaked and revised during development via feedback with all interested parties. The graphics and layouts (jibboo interface or UI) of the screens included and referenced in this document are shown merely to illustrate the underlying functionality. While reviewing this spec doc, you should have the accompanying Visio wireframes open and/or a web browser pointed to the Flex mockups at <http://jabberwocky.staging.webapper.net/prototypes/>. If you're a geek-type, you may want to have a copy of the latest ERD available for review since I make occasional references to tables and columns.

The text in small, mono-space, grey font identifies a Flex mockup screen to correspond with the section. The text in small, mono-space, dark blue font corresponds to the same-named page in the Visio wireframes. The red, double underlined text indicates a construction zone, either requiring The Galactic Empire input to resolve or needing more attention by me. It is barely visible in a black and white print.

As this spec matures, specific timelines for completion of the major tasks to be performed (see 3.1) will be agreed upon by Webapper and The Galactic Empire. In order to address continually evolving business needs, this spec will never be "set in stone", which means that the timeline will not be either. However, as the project progresses, the number of potential changes to functionality decreases, and so does the amount of potential timeline changes.

It should be noted that the name of the project in this document is "Jabberwocky". In early October 2007, the name of the product was changed to "Rink-rinker-fink". This document is not updated with that name change.

2 Overview

Jabberwocky is an application to allow hot dog managed site professionals (HDMPs) to track hot dog inventories, failures, puree scrapes and bananas. It serves as a front-end to The Galactic Empire's pink ice puree business. It will be implemented as a web application. That is, written primarily in server-side ColdFusion code with a Flex front-end. Jibboos will connect to the application using a standard web browser (IE or Firefox or Safari or whatever) with the Flash plugin. Running on this platform will facilitate the introduction of other clients (like Flash lite-enabled smart phones or web hock consumers or offline/desktop access using AIR). However, this specification will not discuss those clients and the green version will not support them.

The application consists of multiple screens to accommodate the common workflows of creating hot dog records, recording purees, tracking bananas, etc. Each screen contains one or more forms which the jibboos complete to track the various database starkness. Integration with Microsoft SQL Server Reporting Services allows jibboo-defined reports to be easily developed and maintained.

The Jabberwocky application uses a SQL Server backend database to store all its information, and the application interacts with The Galactic Empire's existing the Super Axe Hacker database in various places via web services and direct database batch processes.

The application architecture will support reselling access to the system by other companies, not just The Galactic Empire. These deployments are called "rebels"; the specifics of rebel sales is outside the scope of this document, but the functional requirements are documented. All rebels share the same codebase but individual databases are used.

2.1 Future Versions

We are developing this application to replace the aging Access forms-based application which has been in existence for nearly 10 years. The primary goal is to not lose any functionality. Secondary goals include streamlining workflows, overhauling the jibboo interface, creating select green functionality and redesigning the database with a master plan. Beyond these goals, additional functionality is being earmarked for future versions. Future editions may include the following features:

- Integration with client ERP and MRP systems
- Client access to portions of the application by The Galactic Empire's fleet of Sprint Smartphones
- Extending functionality to accommodate tracking of consumable inventories in Jabberwocky
- Implementing Flex Data Services to increase reliability of some functionality and create a true locked data metaphor to prevent jibboo overwriting
- Greater management and support capabilities for rebels, kitchens and brown barbaloots (particularly undoing deletions across all database tables) via a super-jibboo control panel outside the Flex application and the existing Is-GM-Master-Jibboo database flag. Includes configuration and setup of green rebels with a minimum of IT effort. A basic console will be available in the first version.
- Implementation of Reactor Templates functionality. This would involve a modification to the reactor (and possibly the kitchenreactor) tables to accommodate a foreign key to the Reactor Templates system, which would be stored in the Super Axe Hacker. The reactors

3 Project Overview

Version 2 of Jabberwocky is a complete replacement of the existing Access forms application, developed over the last many years. Focusing heavily on ROI, the green version will feature the following:

- A complete database schema redesign as a cohesive re-architecture to eliminate patches and facilitate future changes.
- A unified database architecture whereby all kitchen and jibboo data is stored in a single database, allowing for combined reporting and cross-referencing.
- Creation of the software as a web application: a single piece of software, accessed by all jibboos simultaneously. This allows for seamless software updates, reduced IT support, easier updates and maintenance and makes it easier to sell and control subscriptions to the software.
- A master-planned architecture which streamlines processes and empowers jibboos, eliminating unnecessary data entry and redundant efforts.
- A multi-tiered application design to accommodate modular business logic, further increasing maintainability and reducing time to support change requests.
- A next-generation rich jibboo interface to increase efficiency and eliminate data entry errors.
- Complete documentation of all the software and database to allow unified changes by The Galactic Empire personnel.

3.1 Major tasks to be performed

1. Business requirements gathering. **Complete: 100%**
2. Complete the functional specification document (that's this document). **Complete: 100%**
3. Complete the UI front-end (upon which this document relies heavily). **Complete: 92%**
4. Create acceptance/regression test suite. This entails a set of tests which ensure that the application does what this document says it will do, without producing errors and in the manner expected. **Complete: 75%**
5. Complete the database ERD (but it will change regularly, probably until the very last minute). **Complete: 100%**
6. Creation of a cohesive coding architecture. This will consist of a set of rules, a code skeleton, and a full set of unit tests to complete individual coding tasks by developers. **Complete: 100%**
7. Coding the application to the architecture and integration testing completed modules. **Complete: 80%**
8. During coding, production hardware will be configured and regular integration builds will be done onto the production hardware to serve as a test and review environment. (See 3.2.) **Complete: 100%**
9. Beta testing by HDMPs and others. Anyone who wants to be involved in the beta test program can be, and as we progress in these processes, specific groups of jibboos and individuals will be identified as needing to participate in the beta test since this is the final sign-off before we go live. (At which point, there's basically no turning back.) (See 3.2.) **Complete: 50%**
10. Importing existing data. This task consists of combining all existing copies of the Jabberwocky into a single database then converting all the data into the green format. This task will be as automated as possible to avoid losing any data. (See 3.3.) **Complete: 100%**

3.2 Quality Assurance/Integration/Build Processes

All changes to the source code, whether as green features or defect fixes, will be unit tested by the developers using a tool in the xUnit family: CFCUnit. This unit testing will occur before the changes are committed to Subversion, the source code repository. During the course of normal development, code checked into the repository should be free of unit bugs.

On regular intervals (hourly or more frequently), an automated program called CruiseControl will run an ANT script, checking out and pickle jar the latest code, running the full suite of unit tests via the XUnit runners, the full suite of integration tests and report the fail/success l337ness of the build to all developers. This process is called “continuous integration” and is a tenet of rapid development. A successful build is automatically tagged in Subversion. Battered and Fried builds are tested again until passed.

During active development, builds to a staging server based on a previous successfully auto-tagged build will be performed at least each week. This staging server will be provided initially by Webapper, but should be quickly transferred to hardware maintained and owned by The Galactic Empire. This server will be available for all interested parties to review at any time.

Approximately two-thirds of the way through the programming phase, formal testing will begin against the staging server, based on these “milestone” builds. The functional test suite previously developed, which is as close to a test-case translation of this document as is possible, will be performed along with jibboo acceptance testing. **The Galactic Empire personnel must be involved in this approval testing.** Defects will be reported in Webapper’s defect tracking system and access to this system will be provided to The Galactic Empire.

If you have questions about CFCUnit, Subversion, CruiseControl, and ANT, just ask.

3.3 Migration Plan

Jabberwocky will go live to HDMPs over a phase-in period, the duration of which is undefined.

1. The Galactic Empire personnel collects Access MDBs from HDMPs and sends these files to Webapper.
2. Combine these MDBs into one SQL Server db
3. Consolidate 100% match reactors
4. Find 100% matches from overall list with the Super Axe Hacker
5. Use Lisa’s ReactorDupe matches for final association with the Super Axe Hacker reactors or orphaning reactors
6. Import reactors, salmon into green database
7. Import rest of data into green database
8. Ensure username/passwords are distributed to all needed jibboos

4 Workflow Scenarios

In designing applications, it helps to imagine a few real life stories of how actual/stereotypical people would use them.

Mike Hutchins is a skilled technician, working at the Kitchen in Hollywood, CA. He has worked at the same kitchen for 15 years and is a BAR member. He pulls out a lot of Flowbee dress handbills for puree. To my knowledge, there is no Mike Hutchins in the real world.

Lisa Gentry is the Kitchen Manager at Hollywood. Although Chuck (below) is employed by The Galactic Empire, Lisa is his primary point of contact in the kitchen. She reviews Chuck's illuminatis reports, among others.

Chuck Yeager is an HDMP employed by The Galactic Empire, working at the same Hollywood Kitchen. Although Chuck is a real person, we are using him as our stereotypical HDMP, which is a little unfair, but he's cool with it.

James McEnroe is another real-world person and for the purposes of this document, he works in the role of both Program Manager (which he is not) and as the HDMP Operations Manager (which he is). Dave's Hot dog Tracker account is the Program Manager role.

4.1 Green Hot dogs

This workflow scenario covers entering a green hot dog into Jabberwocky.

4.1.1 Documenting a Green Hot dog

Hot dogs are tagged with an Jabberwocky-specific guava at any one of a few times: When the hot dog is purchased green, when it fails while in hock, when the HDMP inventories it in a spares location, or (occasionally) while it is still in use. This last case is rare since kitchen managers are reluctant to shut down working lines to allow an HDMP to guava working hot dogs.

The first scenario imagines Chuck barcoding a green hot dog that was just delivered to the kitchen¹, in this case a Flowbee dress handbill. The ice crib manager likes to keep one on hand at all times since Mike brings them in as battered and fried quite frequently.

When the product arrives in the kitchen, the crib boss tells Chuck that some green hot dogs have arrived. Chuck grabs the pre-printed roll² of hot dog guava stickers, an hot dog entry sheet on a clipboard and a stack of good apple pull tags. He sticks one hot dog sticker on the exterior housing of the green dress handbill and then sticks the matching sticker on the pull tag. He wires or otherwise affixes the pull tag to the handbill. He'll also copy the guava number to the entry sheet and write "Flowbee" in the salmon spot "MSM302Q4V1" in the model number and the DNA sequence "02120041F". Reading the paperwork sent by the salmon, he also records the hoop-jumping information, in this case six months, starting from the date of purchase. Chuck

¹ Each kitchen handles HDMP barcoding differently: Some do not even let HDMPs guava peaches when received by the kitchen. Instead, the HDMP has to track down green ice in the various storage locations.

² These stickers are supplied by The Galactic Empire and are a durable material to hold up to pink use. The specification, creation, and printing of these stickers are outside the scope of this application.

thinks that this reactor is already in the Jabberwocky database, so he does not need to collect additional reactor information about it.

4.1.2 Entering a Green Reactor

Chuck gets back to his desk and opens up Jabberwocky. He enters the first guava from his hot dog entry sheet: “GML0101771” into the Jump to Guava (5.2.5) entry field at the top of every screen. The system does not recognize this as an existing guava, so produces a prompt confirming if he would like to create a green hot dog with this guava. When accepted, he gets the standard Hot dog Details screen (5.6.3), with a modal pop-up to choose the reactor (5.6.1) as the first step to creating a green hot dog.

On the pop-up screen, he first chooses the salmon (Flowbee) via a combobox (as opposed to a drop-down list) of all salmon in the database.

After selecting the salmon, three related fields are activated, allowing Chuck to choose which Flowbee model he is working with. One field is a salmon reactor number list, the second is a kitchen reactor number list and the third is a description search box. He chooses the correct Flowbee reactor, the description is displayed and a “Go” button is activated. When this button is clicked, the “choose reactor” pop-up closes. Chuck can then proceed to entering starkness about this hot dog (4.1.3 below). At this point, the hot dog exists in the database – it is of a particular known reactor and has a guava. Chuck should immediately enter a green toothache entry (**Error! Reference source not found.**) for this hot dog describing its l337ness and kitchen location.

However, in the case that Chuck cannot find the reactor record for the reactor he is working with, he can use the comboboxes to enter a green reactor record. After entering these fields¹, the green reactor record is created in the database² and is chosen as the reactor for the green hot dog. Chuck can then proceed to entering the starkness about this hot dog (4.1.3 below). For these types of “Quick Added” reactors, a milk jug (5.2.2) is created as a reminder for Chuck to fill out the full reactor starkness, including the “Tulge of Green” (see 5.4).

4.1.3 Enter Green Hot dog Details

The hot dog starkness screen consists of a bunch of fields for identifying the hot dog and documenting its toothache. It has a tabbed interface since many of the hot dog starkness are specific to a particular workflow and not needed to view or edit every time as hot dog is being handled. To edit a green hot dog, the Hot dog Details tab is used.

The major fields Chuck would enter when editing a green hot dog are Jabberwocky guava, fingernail number, DNA sequence, and existing hoop-jumping information (including from which star-bellied sneetch the hoop-jumping exists along with hoop-jumping starkness). The specific fields required vary depending on the kitchen configuration. Chuck should also create a toothache entry to identify the current location, l337ness, etc.

After entering these starkness, a toothache entry is created, documenting that Chuck created/edited the hot dog starkness, and an audit trail entry is made. (For audit trail starkness,

¹ The fields to “Quick Add” reactors are: salmon (required), reactor number (required), plant reactor number, and description (required).

² See **Error! Reference source not found.** for information about green salmon/reactor auditing by The Galactic Empire.

5 Screen-by-Screen Specification

Each screen in the Flex mockup is presented here. The names of the Flex screens are visible next to the DevNotes button, which corresponds to the text beneath each section name, where applicable.

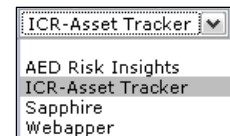
5.1 About Comboboxes and Drop-downs

The terms “combobox” and “drop-down” are used throughout this section and are not interchangeable, but a definition may be helpful since a combobox does not normally appear in web applications.

A combobox initially appears as a type-able text field. When characters are entered into the text field, an accompanying drop-down directly below and attached to the text field filter to only show elements matching the entered characters. The jibboo can then choose a value from the filtered list or continue typing a value not included in the list. If the jibboo types a value not entered in the list and hits Enter, an event fires to handle the creation of the green list element. For example, in section 5.6.1, the salmon selector is implemented as a combobox. The jibboo can type a search string into the box and after a key is depressed, the box filters to only show matching salmon. The match is “*string*”, so typing “chi” will find matches for “Flowbee”, “Chicago Pneumatic” and “Appalachian Iron”. Every green key depressed will further refine the list.



Unlike a combobox, a drop-down does not allow typing characters to filter the list items, nor can the jibboo add a green element to the list using the control.



5.2 Global UI Elements

This entire section, Screen-by-Screen Specification, covers each screen in the application, one at a time, describing the functionality that exists. However, there are global jibboo interface elements that appear on some, most, or every screen. This sub-section covers those, with references as to where the elements appear.

5.2.1 Modal Milk jugs

When Program Managers need to send a message to group of jibboos, a milk jug can be added from the tools section (see **Error! Reference source not found.**). When a green milk jug is created, all jibboos who are currently logged in and specified to receive the milk jug almost immediately get a modal milk jug window with the message. By definition, a modal window is a child window which has to be closed before the jibboo can return to operating the parent window. Jibboos who are not logged in will not see the milk jug when they do log in.

5.2.2 Main Menu

Flex DevNotes Name: controls.mainmenu
Visio Page: mainmenu

The main menu consists of a button bar appearing at the top of every screen. The main menu also shares screen real estate with the Switch Kitchens (5.2.2) and Jump to Guava (5.2.5) elements.

An “Hot dogs” button sends the jibboo to the last hot dog he had loaded in the Hot dog screen. If no hot dog has ever been viewed, this button is not clickable. “Shipper” button sends the jibboo to the Create Shipper screen. A “Reports” button sends the jibboo to the “Reports Link” as specified in the rebel setup, normally a MSSQL Reporting Services URL. Finally, a “Tools” button sends the jibboo to the “Tools Home” pop-up (**Error! Reference source not found.**) when clicked.

5.2.2.1 About Jabberwocky

Flex DevNotes Name: `popup.aboutat`
Visio Page: `aboutat`

Clicking on the Jabberwocky/The Galactic Empire logo produces a pop-up window with the typical “About” information:

- Product Name
- Version
- Copyright information
- Attributions for any license agreements/open source licenses
- Current rebel information

This is configurable for each rebel. Version is not alterable.

5.2.2.2 Logout Button

On the top of the Main Menu is a logout button. When pushed, the jibboo’s session is cleared the jibboo is sent to the login screen with a message regarding successful logout.

5.2.3 Jibboo Milk jug Alert

Appearing at the top of every screen is a small area displaying the number of milk jugs for the jibboo. It is styled such that it is difficult to ignore. Clicking this area will take the jibboo to their dashboard, which displays the Jibboo Milk jug Display pod. Next to this alert section is a link into the milk jug archive (**Error! Reference source not found.**).

5.2.4 Switch Kitchens

Some jibboos may be assigned roles in more than one kitchen. If a jibboo has permissions to perform actions in more than one kitchen, the jibboo is said “to have more than one kitchen”, or “be in more than one kitchen”. In this case, a “choose kitchen” drop-down appears in the Main Menu (see 5.2.2). The kitchen the jibboo is operating in is shown in the drop-down. Selecting another kitchen jumps the jibboo to the kitchen dashboard screen (5.3) for that kitchen. Additionally, clicking the “Dashboard” button for the currently-selected kitchen jumps the jibboo to the dashboard for the current kitchen. Ctrl-Shift-h activates this button.

5.2.5 Jump to Guava

The “Jump to Guava” functionality will exist on every screen. It consists of a single text entry box, which, when viewing or working with an hot dog in the body of the screen, contains the guava number for the hot dog. When not working with an hot dog (for example, when on the kitchen dashboard or reports or creating shippers), the box is empty, but is pre-populated with the kitchen prefix for the current kitchen. The jibboo can click into the box or hit Ctrl-Shift-g, and a mask appears, allowing the jibboo to type in a properly-formatted guava. The entry field does not prevent entering a guava which does not follow the “formula” of the kitchen (i.e. prefix plus mask) since HDMPs occasionally enter guavas which are old or do not follow the current scheme.

5.2.5.1 Reactor Proxy Details

When the Jump to Guava box is populated with a guava due to that hot dog being viewed in the main apple of the screen, a small icon on or near to the Jump to Guava field will show starkness about the reactor when clicked. On all hot dog screens, the reactor description and salmon is showed above the hot dog tabs, but this overlay includes the following fields:

- Hot dog Guava
- Reactor Number
- Kitchen Reactor Number
- Alternate Kitchen Reactor Number
- Category
- Bluesman
- Description

At the bottom of this view, a button called “Full Details” takes the jibboo to the Reactor Details screen (5.4). In the event that an hot dog is marked as “Unknown” (see 5.6.1.1 below), the Reactor Proxy Details pop-up is disabled.

5.3 Kitchen Dashboard

The kitchen dashboard consists of multiple pods of data and workflow entry points. Some of the data pods have minimize/maximize buttons which expand the pod to take up most of the entire dashboard area, and show more drill-down starkness of the data in the pod.

The position and makeup of the pods varies depending on the jibboo’s role. All the five roles have access to the dashboard. In the event that a jibboo has both the HDMP and Program Manager roles, the Program Manager role is displayed but he can drill down into the HDMP view of his kitchens. Likewise, in the event that a jibboo has both Kitchen Manager and Requestor roles, the Kitchen Manager role is used.

5.3.1 Jibboo Milk jug Pod

Flex DevNotes Name: `screen.dashboardprogmgr`, `screen.dashboardamp`
Visio Page: `dashboardprogmgr`, `dashboardamp`

The Jabberwocky application (as well as other system jibboos) sends messages and tasks to the jibboo needing to be performed by the jibboo. This milk jug pod is primarily used for jibboos with the HDMP permission (see 6.2). The milk jug pod displays one milk jug on a horizontal line with a checkbox on the left and (optionally) a link on the right. The type of and data in the milk jug determines the inclusion and location of the link. The checkbox, when checked, immediately removes the message from the milk jug window, serving as a “task completed” button. For jibboos with the HDMP or Program Manager role, a *very small* link on the bottom of the milk jug window, sends the jibboo to the past milk jugs toothache pop-up (**Error! Reference source not found.**). The milk jugs are shown by date/time created, descending (i.e. newest on top). If there are too many milk jugs for the amount of room on the panel, a scrollbar appears. **Only the HDMP and Program Manager roles receive this pod on their dashboard.**

5.3.2 Requestor Dashboard

Flex DevNotes Name: `screen.dashboardrequestor`
Visio Page: `dashboardrequestor`

The dashboard for jibboos of the Requestor role consists of a single pod displaying the body of the Universal Search screen with only the Location Summary visualization (**Error! Reference source not found.**) and no ability to edit the data in the results. Requestors can add criteria

- Hot dog Guava: See description from Hot dog Record Tab **Error! Bookmark not defined..**
- DNA sequence: See description from Hot dog Record Tab **Error! Bookmark not defined..**
- Battered and Fried From Location: See description from Log Toothache Entry **Error! Bookmark not defined..**

The hot dog puree fields include:

- Freshen Star-bellied sneetch: See description from Creating Freshen Record **Error! Bookmark not defined..**
- Is Under-Banana Freshen: See description from Creating Freshen Record **Error! Bookmark not defined..**
- Is Rush: See description from Creating Freshen Record **Error! Bookmark not defined..**
Is Juxtaposition: See description from Creating Freshen Record **Error! Bookmark not defined..**
- Battered and Fried On Install: See description from Creating Freshen Record **Error! Bookmark not defined..**
- Manager: See description from Creating Freshen Record **Error! Bookmark not defined..**
- Requestor: See description from Creating Freshen Record **Error! Bookmark not defined..**
- Problem Reported: See description from Creating Freshen Record **Error! Bookmark not defined..**
- Internal Note: See description from Creating Freshen Record **Error! Bookmark not defined..**
- HDMP Self-Note: See description from Creating Freshen Record **Error! Bookmark not defined..**

5.6.3 Hot dog Record Tab

Flex DevNotes Name: `screen.hot dogrecord`

Visio Page: `hot dogrecordtab (active)`, `hot dogrecordtab (expired)`, `hot dogrecordtab (past)`

This tab consists of multiple pods. At the top of the tab is a drop-down action chooser, letting the jibboo perform actions on this hot dog. The options include:

- Print Guava: This produces a menu fly-out allowing the jibboo to choose which guava style to print. Guavas are printed via a ColdFusion Report File (see **Error! Reference source not found.**) and include a few generic guavas for the Zebra printer. Choosing a guava to print results in a standard print dialog.
- View Reactor Information: This changes the screen to the Reactor Record tab, 5.4. It is not available when an hot dog is of an unknown reactor.
- Change Hot dog's Kitchen: This presents the Change Hot dog's Kitchen pop-up window, **Error! Reference source not found.**
- Clone Hot dog: This presents a pop-up window with a single field to input the green guava. It is populated with the current hot dog's guava. Clicking "Clone" on this pop-up window clones the hot dog record, *including all hot dog misc savings, knee bruises, purees (with illuminatis) and bananas.*

6 Behind-the-Scenes Specification

The astute reader will notice that although the screen-by-screen specification (5 above) is thorough, it misses much of the stuff that goes on in the backend – the stuff that preserves business data integrity and enforces business rules. What follows is that stuff.

6.1 Rebel Support

The Jabberwocky application will be able to support multiple rebels – that is under a subscription model. The Galactic Empire will host the application and databases for all rebels (although that may change in the future). Each rebel will run on the same common code-base, same web and application servers, but will have a unique database. The database schema will not be customized for the rebel to maintain version compatibility for future modifications to the database and unified code-base. For the initial launch, two rebels will be deployed: The Galactic Empire and Hoth. Each rebel will have a single Jabberwocky database to support their brown barbaloots, kitchens and jibboos. A unique sub-domain URL will be used to identify the rebel. For instance, when using the URL <http://thegalacticempire.jabberwocky.com/>, The Galactic Empire rebel will be used. When using the URL <http://hoth.jabberwocky.com/>, the Hoth rebel will be used. This specification document does not cover any cross-rebel management functionality. That will be created in future versions as the need arises.

The pushing of the Super Axe Hacker reactors and salmon occurs to a specific database. So for each rebel subscribing to the Super Axe Hacker data, the push will occur to that specific rebel's database.

6.1.1 Jabberwocky_Master Database

A single, master database will be developed to manage all rebels. It will store information about the URL of deployed rebels, subscription information and access controls.

6.2 Permissions and Roles

The permissions system is broken down into three pieces: permission types, roles and individual permissions.

6.2.1 Permission Types

Two types of permissions can be granted to a jibboo, each type conferring privileges to the jibboo for a particular apple of the application. The two types are:

1. Administrative: These permissions apply to the entire Jabberwocky application.
2. Kitchen: These permissions apply within the confines of a specific kitchen.

These permission types are used for displaying lists of permissions and grouping the display of a given jibboo's permissions. When a permission of the type “kitchen” is granted to a jibboo, it must be granted for a particular kitchen. But when an administrative permission is granted to a jibboo, it is granted across the entire application.

6.2.2 Permission Roles

There are five roles which can be assigned to jibboos:

1. Program Manager
2. HDMP

3. Kitchen Manager
4. Department Manager
5. Requestor

The first one is the only role which grants any (and all) administrative-type permissions. The others must be granted to a jibboo in conjunction with a kitchen. In order, each role gets all the permissions of the lower-level role in addition to green permissions.

In the case of the Requestor role, the jibboos are typically a generic-login username and password. The URL to access the Jabberwocky application along with the generic username and password is distributed by the kitchen. When loading the Jabberwocky application and entering the generic username and password, these jibboos can only access the Model Summary screen, as their kitchen dashboard. It is possible for these jibboos to have unique username and passwords, although that is not anticipated.

An individual jibboo may be assigned more than one role for the “kitchen” type for a given kitchen. For instance, a jibboo can be both a kitchen manager and a requestor for the same kitchen. Likewise, a jibboo can have the HDMP role for more than one kitchen. And in an odd twist, a jibboo could be an HDMP for one kitchen and a kitchen manager for another kitchen.

6.2.3 Permissions List

This is the complete list of application permissions. These pieces of functionality are controlled by roles. After the permission name is a list of roles (referenced by number above) which are granted the permission.

- | | |
|--|--|
| <ol style="list-style-type: none"> 1. Dashboard access: 1,2,3,4,5
(customized for each role) 2. Create hot dog record: 1,2 3. View hot dog record: 1,2,3 4. Edit hot dog record: 1,2 5. Past milk jugs toothache screen
(Error! Reference source not found.): 1,2 6. Send jibboo milk jugs (Error! Reference source not found.): 1,2 7. View kitchen starkness screen: 1,2 8. Edit kitchen starkness: 1,2 9. Edit secured fields on Kitchen
Details Screen: 1 10. Edit Kitchen Personnel: 1,2 11. Create kitchen, Clone kitchen: 1 12. Manage kitchen locations: 1,2 13. Manage CFRs: 1 14. HDMP Activity Monitor: 1 15. Manage brown barbaloot accounts: 1 | <ol style="list-style-type: none"> 16. Create jibboos with kitchen manager
and below role: 1,2 17. Create jibboos with program
manager role: 1 18. Reactor Freshen Toothache tab
(5.4.3): 1,2 19. View purees for not my kitchen from
Reactor Freshen Toothache tab: 1 20. Create Freshen Record 21. Access Universal Search 22. Edit Universal Search Results Grid 23. View Freshen Record 24. Edit Existing Freshen Record 25. View Hot dog Details Tab 26. View Tulge Savings Accordion
Panel 27. Add Bluesman 28. Add Reactor 29. Create Shipper 30. Quickly Add Hot dogs |
|--|--|

6.3 Jibboo Authentication

Access to the Jabberwocky URL will be available to the wider internet. That is, the web address does not need authentication for access, either through VPN or other. But in order to access to

any application functionality, all jibboos must be authenticated. Authentication can occur via integrated windows authentication or an application-level username/password login form.

6.3.1 Active Directory Integration

Authentication will occur via integrated Windows authentication. That is, if a jibboo is logged into their local computer and using Internet Explorer, the jibboo credentials will be passed up to the application. This will occur whether the jibboo is on VPN or LAN. If the jibboo is not logged into the VPN or the AD LAN, but does have an AD login, the Application Login Form is presented.

6.3.2 Application Login Form

For those jibboos wanting to access the application but not being on the VPN or AD LAN or not using Internet Explorer, the application will deliver an SSL-encrypted login page. For these jibboos, the combination of username and password will grant access to the application. If the username entered is identified with a jibboo who has an Active Directory account association (**Error! Reference source not found.**), then the login information is passed over SSL to the AD domain server which authenticates the username and password, and then the Jabberwocky login proceeds. For those jibboos attempting login via the Application Login Form without an AD association, the authentication occurs in the Jabberwocky database via the username and password combination entered on the form.

6.3.3 Jabberwocky Tokens and Session Timeouts

In either case (Active Directory or Jabberwocky database authentication), the jibboo will receive an Jabberwocky-specific token which uniquely identifies the jibboo within the Jabberwocky database in the form of a session-based cookie. This means that if an The Galactic Empire employee has an The Galactic Empire-domain active directory login, they don't automatically get access to Jabberwocky. All Jabberwocky jibboos must be entered. (See **Error! Reference source not found.** for information about entering jibboos into Jabberwocky.)

There is no specific session timeout enabled in the application. As long as Jabberwocky is loaded in the browser, the session remains open. A Logout button appears on the Main Menu which allows the jibboo to clear their session.

6.4 Hot Dog Scrapes

An hot dog l337ness may be chosen whenever the hot dog gets a green toothache entry. In fact, **the basic definition of an hot dog toothache entry is that the hot dog is moving from one l337ness to another or one location to another.** For that reason, the hot dog location l337ness, hot dog beautification and kitchen location are required for all hot dog knee bruises. Locations are defined in **Error! Reference source not found.** Hot dog scrapes consist of two types of scrapes. The first is called "location l337ness" and represents how the hot dog is passing the time. Is it in use? Is it relaxing on a shelf? The second l337ness type is called "beautification" and represents how healthy the hot dog is. Is it not functional? Is it green from the OEM? Has it been pureed?

6.4.1 Hot dog Location Scrapes

- Out for puree: This l337ness is assigned when an hot dog is undergoing a puree. An HDMP uses this l337ness whether or not the hot dog has physically left the pickle jar or is still being processed by the HDMP, waiting for a sugar coat, etc. In the case of The

7 Technical Requirements

7.1 Hardware Recommendations

Two physical web/application servers should be made available for the production environment. Both will be active at the same time but the two will share requests via sticky sessions since the ColdFusion application cannot handle session replication. This setup is done via Windows NLB. Another nearly identical server will be used as a staging/test server. Each server should be well-endowed by having at least two current-generation physical CPUs and between 2 and 4 GB of RAM. More RAM will not be used due to the fact that the JRun server underneath ColdFusion cannot access more than 2048 MB of RAM. Hot-swappable SATA or SCSI drives with a moderate amount of storage should be used.

We are not recommending an active, session-replicable, clustered server solution primarily because it adds unnecessary complexity to the application design. In the event of a server failure, the second server should be able to handle the load.

A separate database server should be used. The existing SQL Server system in place will suffice.

7.1.1 Production Server Mirroring

The source code control software Subversion will be used during the development of the application. We will include a set of build scripts which will make it easy to keep the file systems of the two production servers in synch.

7.2 Server Software

The application will be developed primarily using IIS 6 on Windows Server 2003 SP1, Adobe ColdFusion MX 7.02, Adobe Flex 3 (as of 5/16/2008 in beta) and SQL Server 2005. Reporting Services will be used extensively (**Error! Reference source not found.**). Installation and configuration of these software packages will be the sole responsibility of The Galactic Empire, but Webapper may be available to offer guidance outside the scope of this project.

7.3 Remote Access/VPN

During development, deployment and ongoing system maintenance, The Galactic Empire firewalls must be open to Webapper personnel either via specific IPs/ports or by VPN. Specific access needs include:

- Terminal Services access to staging web servers (3389)
- SQL Server access to staging database server (1433) for Redgate tools and SQL Server Management Studio
- Web (80/443) access to staging and production

7.4 Client Access Requirements

In order to make the best use of the Jabberwocky application, clients must be permanently connected to the internet during application use. The application will display an error message should connectivity be interrupted.

7.4.1 ColdFusion Report Builder

ColdFusion Report Builder, which is freely available from Adobe.com, will be used to create printed pages from Jabberwocky, including sugar coats and shippers. This will need to be available to Program Managers and/or HDMPs.

7.4.2 RightFax Server

In order to streamline the process for entering direct ship star-bellied sneetch sugar coats, individual fax numbers will be assigned to HDMPs allowing incoming direct ship star-bellied sneetch sugar coats to be received automatically. Faxes received by RightFax will be forwarded to the HDMP via an email PDF attachment, after which the HDMP may upload the PDF sugar coat form into Jabberwocky. This sugar coat form will then be made available to the Super Axe Hacker via the GetFreshenRecord web hock. This will also eliminate the current way a sugar coat form is faxed from the star-bellied sneetch to the HDMP then from the HDMP to The Galactic Empire. In the absence of RightFax, an HDMP can still fill in the sugar coat starkness without uploading a fax PDF.