BLIS Developer Notes

Module1: Authentication

1. Users/login.php: This is the login page.

Bugs:

Change the tip from:

If you have forgotten your password then please send an email to 'c4gbackup@gmail.com' with the subject 'Password'.<br> New password will be sent to you.

To:

If you have forgotten your password then please contact your BLIS admin.

If the username or password is incorrect, the username textbox must not get emptied.

2. Users/validate.php: This validates user credentials passed by login.php and sets the user session variables.

Bugs:

Hardcoded value 17 on line 31.

Module: One time password reset

*Vulnerable Must be eliminated*

Includes/password_reset_need.php
Users/ oneTime_password_reset.php
Ajax/ oneTime_password_reset_confirm.php

Module 3: Authorization
Includes/user_lib.php: Contains constants for user roles and logic for choosing top menu.
Includes/perms_check.php: Checks if user is logged in and fetches top menu.
users/lab_user_new.php: For creating new user.

users/lab_user_type_new.php: Add a user type/role.
users/lab_user_type_edit.php
users/ lab_user_edit.php
ajax/lab_user_delete.php: Deletes the user for given id.
ajax/lab_user_type_delete.php: Deletes the given user type.
Includes/page_elems.php: functions getlabuserstable and getlabusertypes create the UI for the user account option.
Config/lab_config_home.php: Has the menu option for user accounts.

BLIS supports different types of user roles and each of these roles can access different modules under BLIS.

Following default roles are available:

Role1: Technician read-only

Role2: Admin
Role3: Superadmin
Role4: Country Director
Role5: Clerk
Role 6: Technician who can only see reports.
Role 7: Technician who can only add results.
Role 8: Technician who can see reports and add results.
Role 9: Technician who can only do registration.
Role 10: Technician who can do registration and see reports.
Role 11: Technician who can register and add results.
Role 12: Technician who can do registration, add results and view reports.
Role 13: Technician that can see patient name.
Role 14: Technician read-write
Role 15: Technician who can verify
Role 16: Read only mode.
Role 17: Physician

Custom roles can be added by lab admin.

The role definitions or “user types” are stored under the table user_type in the blis_revamp database.

Constants for all the above roles are defined in includes/user_lib.php, any new roles must be added here and only constants must be used in code.

The role-based access control limits access to following modules:

| ID | Name | Code File |
|---|---|---|
| 2 | Patient Registration | find_patient.php |
| 3 | Test Results | results_entry.php |
| 4 | Search | search.php |
| 5 | Reports | reports.php |
| 6 | Inventory | view_stock.php |
| 7 | Backup Data | backupDataUI.php |
| 8 | Lab Config | Lab_configs.php |
| 9 | Catalog | Catalog.php |
| 10 | Lab Admins | Lab_admins.php |
| 11 | Catalog | Country_catalog.php |

| Role ID | Page ID |
|---|---|

| | |
|---|---|
| 16 | 5 |
| 17 | 2,3,4,6,7 |
| 5 | 2,4 |
| 13,14 | 2,3,4,5,6,7 |
| 1 | 4,5 |
| 2 | 5,7,8,9 |
| 3,4 | 8,10,11 |
| 15 | 3 |

BUGS:-

1. Access control does not work, if I assign the user role of 001 which should give only reports access, still user is able to access everything. Similarly checked for tech_ro.
2. The rwoptions must be linked to user roles. I find the functionality being duplicated while adding user accounts as well as while adding user types.
3. The reports check box is missing from the list of options i.e. the 4 writable options can be chosen and everyone has access to reports by default.
4. The password textbox is not masked while creating new user account.
5. Show patient name can be selected with any other role but in code it is treated as a different role.
6. The logic for deciding menu for has unreachable code for custom roles.

Module 4: Updates

BLIS update happens in 2 stages. The first step involves copying the files from the BLIS update archive to the BLIS installation directory and running the update batch file. The second phase requires the BLIS admin to install the update for their lab by following the update link on the home page.

Associated code files:

Users/Home.php: Checks if the update has been applied, if no, displays the update section with link to complete the update.

Includes/defaults.php: Contains constants used in the blis system including $VERSION which denotes the current version of BLIS in use.

Update/check_version.php: Checks if an entry exists in version_data table for the current version in global variable.

Associated Database table:

Version_data

Preparing the update manually:

1. Change the value of $VERSION in defaults.php to the new version number.
2. On line 31 of update/blis_update.php, add the new version number to the $version_list array.
3. Add the filename of the admin script to $version_doc_admin in blis_update.php. The corresponding .sql file must exists in htdocs/data folder. This script will be run on the blis_revamp database when update is performed by BLIS super admin or country director.
4. Add the filename of the script to be run on the blis_revamp database to $version_doc_revamp in blis_update.php. The corresponding .sql file must exist in htdocs/data folder.
5. The filename of the script to be run on the specific lab database must be appended to $version_doc_lab in blis_update.php. This script is run on blis_<lab_id> database. The lab id is determined by the lab to which the user performing the update is associated with.
6. Add the updated language files to htdocs/Language folder. These files should be one of the following:

a. Default.php and default.xml – for default language setting
b. En.php and en.xml – for English.
c. Fr.php and fr.xml – for French.

7. To prepare the update zip:-

a. Zip the htdocs folder to htdocs.zip.
b. Create a new directory to hold all the update files. It must contain following:

- Htdocs.zip from above step.
- 7za.exe: Used for unzip.
- XXMKLINK.exe: To create desktop shortcut.
- md5.exe: For checksum check.
- Gdiplus.dll: Dependency library.
- Feature_List.txt: List of features added in this update.
- Splash.png: The icon for new blis version.
- update_C4GBLIS_v<version_number>.bat: See below contents.

- Update_Instructions.txt: See below for contents.

c. Create a windows batch file and call it update_C4GBLIS_v<version_number>.bat

d. Put the following in it, replace <version_number> with blis update version number.

- <u>Guide on language flexibility:</u>
  - Add the key value pair for the language agnostic words/sentences to default.php, default.xml, en.php, en.xml, fr.php, fr.xml files in the htdocs/Language folder.
  - These files shall be moved to all /local/langdata_* folders upon update. This part of copying Language files to local folder happens upon clicking the “Click here to complete update to version x.x” button that appears on the homescreen after running the update bat file (i.e the copying doesn’t happen in the bat script). Code relevant to this can be found in blis_update.php and a function “update_language_files” in db_lib.php.
  - While running C4G Blis in local, the language flexible sentence mappings are fetched from local/langdata_* (there are files for each lab id), and so if we only add the language changes to the files in htdocs/Language those changes will not be reflected while the developer is testing it. One way to overcome this is make the changes in the corresponding local/langdata_* folder as well so that they can be tested while developing.

- Things to keep in mind regarding the version update while dev testing:
    - Once you have updated to a version number x, if at a later point of time you try to remove that instance of C4G Blis and update an older version to the same version number x, BLIS will not show you the “Click here to complete update to version x” and so the update will be incomplete.
    - This is because whenever Blis is deployed it checks whether the current version number is present in the ‘version_data’ table and if not then it shows the ‘Click here to complete update to version x’ button and upon clicking this button this version number is added to the ‘version_data’ table so that later on this button isn’t showed.
    - Because of this if we try to update to the same version number again while dev testing, this button won’t be shown and so the update will be incomplete.
    - One way to overcome this is to manually edit the code where the check whether to show the said update button happens (i.e in home.php) and make the condition true, complete the update and then revert that change back. Note that this is only for developers while testing. Ideally users will not have to update to the same version twice.

```
@echo ON

For /f "tokens=2-4 delims=/ " %%a in ('date /t') do (set mydate=%%c-%%a-%%b)

For /f "tokens=1-2 delims=/:" %%a in ("%TIME%") do (set mytime=%%a%%b)

set ts="%mydate%_%mytime%"

:checksumcheck

        echo Updating to C4G BLIS v<version_number>... Please wait...

        if not exist htdocs.zip goto failurezip

        if not exist md5.exe goto failuremd

        md5 -2BC35F41B66A6338EFE2FFBCD8B71588   htdocs.zip

        IF ERRORLEVEL 1 goto checksumfailure

        IF ERRORLEVEL 0 goto checksumsuccess

        goto checkfailure
```

```
:updation
	if not exist 7za.exe goto failureza
	echo 7za.exe file exists
	echo Backing up the present version of BLIS...
	ren htdocs htdocs_backup_%ts%
	if "%errorlevel%"=="1" goto failurefolder
	echo Backup complete.
	echo .

	echo Starting C4G BLIS v3.41 source code update...
	echo Extracting htdocs.zip...
	7za x htdocs.zip -aoa>tmpFile2
	if "%errorlevel%"=="1" goto extractionfailure
	del tmpFile2
	echo Code update complete.
	goto success

:failuremd
	echo md5.exe not found!
	echo BLIS Update failed!
	goto eof

:checksumsuccess
```

```
        echo Update file verified to be correct.
        goto updation

:checksumfailure
        echo htdocs.zip file is corrupt!
        echo Redownload the update files and start the update process again.
        goto eof

:success
        echo Creating Desktop Shortcut
        xxmklink "C:\Users\%username%\Desktop\BLIS.lnk" "%cd%\BLIS.exe" "" "%cd%"
        echo C4G BLIS updated Successfully to v<version_number>!
        echo Starting C4G BLIS v<version_number> ...
        start BLIS.exe
        goto eofExit

:failurezip
        echo htdocs.zip not found!
        echo BLIS Update failed!
        goto eof

:failureza
        echo 7za.exe not found!
        echo BLIS Update failed!
```

```
        goto eof

:failurefolder
        echo htdocs folder not found!
        echo BLIS Update Failed!
        goto eof



:checkfailure
        if exist del tmpFile2
        echo Error occurred during file verification
        echo BLIS Update Failed!
        echo Try again.
        echo If error persists, email m.dhruvchand@gmail.com to report this error.
        goto eof

:extractionfailure
        if exist del tmpFile2
        echo Error occurred during file extraction
        echo BLIS Update Failed!
        echo Try again.
        echo If error persists, email m.dhruvchand@gmail.com to report this error.
        goto eof
```

```
:eof
        if exist del tmpFile
        if exist del tmpFile2
        ping -n 100 -w 1000 0.0.0.1 > NUL

:eofExit
        if exist del tmpFile
        if exist del tmpFile2
        exit
```

e. Following are the update instructions:

- Instructions on how to perform the update are as follows,

1. Download and Extract the upgrade file.

2. Open the folder where you extracted the update files and Copy all the files inside and replace them to your C4G BLIS folder.

3. Run update_C4GBLIS_v<version_number>.bat to start the update process.

4. Wait for C4G BLIS to startup. Login and click on the update button on the home page.

5. Once the update completes, you can continue to use the system.

Bugs/ Fixes:

1. Default currency setting and exchange rate is ambiguous on Line 91 of blis_update.php.
2. Remove all the extra files from /update folder apart from blis_update.php, check_version.php and redirect.php.
3. MD5 verification in batch file is not working for update.

Module 5: User Profile

The following attributes are stored in the BLIS system for a user:

1. Username: Cannot be changed once created.
2. Name: User's full name.
3. Email: User's email address.
4. Phone: Phone number.
5. Language: Prefered language English/ French/ default.
6. User password.

Associated DB Table: user

Associated code files:

1. Users/edit_profile.php: The UI for edit profile and change password forms.

Bugs:

By default one of the forms must be displayed out of change password or edit profile.

Minimum password length is 3, should be larger and password complexity must be validated.

Validations missing for email and phone.

2. Users/change_profile.php: Performs update into the database based on user supplied values.

3. Users/change_pwd.php: Checks if the old password matches current password and updates the new password in DB.

Module 6: User Feedback

User feedback is asked for upon logout from the BLIS system.

Associated DB Table: user_feedbacck

Associated Code Files:

1. Feedback/user_rating.php: The user interface for taking user feedback.

Bugs:

Hard coded for English and French. Strings must be added to language files and redundancy in UI must be removed.

2. Feedback/user_rating_submit.php: Inserts user feedback in DB.

Bugs:

Logic must be pushed to db_lib.php to make it standard.