# “Bonjour”
# **Defect Log**

**Last Revised:** March 7, 2014


**Team 7 Members:**

Xiangyu Bu

Rishabh Mittal

Yik Fei Wong

Yudong Yang

# Introduction

In this document, the first section introduces how to set up the environment for the project, and how to install and run the program. The second section is a list of 50 bugs that we suggest to catch.

# Installation and Configuration

## Setting up the Server

The following content introduces how to set up the server environment and have the program run.

### Step 1: Setting up the Environment

Although the server code works on Windows + Apache + MySQL + PHP (WAMP) environment, we assume to set up a typical environment for PHP: Linux (Ubuntu) + Apache + MySQL + PHP. The necessary components and their versions in this environment are:

| Component | Version | Note | Replacement |
|---|---|---|---|
| Ubuntu | 13.10 | The operating system | Microsoft Windows |
| PHP | 5.5.8 | The scripting engine. Either newest PHP5-MPM package or newest PHP5-FPM package should work. | None. |
| Apache | 2.4 | Server http daemon | Nginx 1.5.8 |
| MySQL | 5.6 | Database | MySQL 5.5 |
| PHP extensions | - | php5-curl php5-gd php5-intl php-pear php5-imagick php5-imap php5-mcrypt php5-ming php5-ps php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl opcache enabled. | |

An article of setting up Ubuntu + Nginx + PHP-FPM + MySQL can be found here: http://xybu.me/setting-up-a-ubuntu-server/

If Microsoft Windows is used, make sure set up smtp configuration correctly to send emails.

**Step 2: Getting Source Code**

## Installing Client App

# List of Suggested Defects

## Server-side Defects

| #1 | Server has been hosted to a Raspberry Pi device. | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | The server code is hosted on mc18.cs.purdue.edu cluster, which is a high-performance server. The virtual host has large memory and high-performance CPU. | | | | |
| Output after seeding | Raspberry Pi has a 700MHz ARM11 CPU and 512MB memory, which is significantly lower than mc18 server. This difference in performance is large enough to fail volume tests and stress tests. | | | | |
| Suggested Correction | Use a local computer or virtual host to host the server program, or contact Bonjour team to access the original hosting. | | | | |

| #2 | User profile can contain code that allows for XSS attack. | Type | **Black box** | Severity | **2** |
|---|---|---|---|---|---|
| Output before seeding | The function `filterHtml` in `Core` class will filter out any HTML or Javascript code contained in the fields submitted. So the code will be shown as text and will not be executed. | | | | |
| Output after seeding | The user profile fields will have the original text sent by the sender (including a faker program). | | | | |
| Suggested Correction | In include/Class.Core.php, finish the filterHtml function to filter out special characters like "&" to "`&`", ">" to "`>`", "<" to "`<`", etc. | | | | |

| #3 | Array out-of-bound exception if birthday field does not conform to the predefined format. | Type | **White box** | Severity | **2** |
|---|---|---|---|---|---|
| Output before seeding | When a profile field is named "`birthday`", program split the value with delimiter "-", and the string "`MM-dd-YYYY`" is made three parts "`MM`", "`dd`", "YYYY". If the size of the array is not three, report error. If any part is not | | | | |

| | |
|---|---|
| | numerical value, return error. And at last, check if the three fields form a valid date. |
| Output after seeding | After seeding, the program will not check the array size, but will still assume there are three elements in the splitted array. Thus, if the birthday sent is of format "`mm-YYYY`", accessing `array[2]` will yield error. |
| Suggested Correction | In `action.php`, check the size of the splitted array of `birthday` field, and return error if the size is not `3`. |

| **#4** | The `sendEmail` function is disabled. | Type | **Black box** | Severity | **3** |
|---|---|---|---|---|---|
| Output before seeding | The `sendEmail` function in `Core` class is a wrapper of several Linux sendmail binaries. It will choose a proper one to handle PHP send email requests. | | | | |
| Output after seeding | No email will be sent regardless of the situation. | | | | |
| Suggested Correction | In the function body of `sendEmail`, redirect the arguments to php built-in `sendmail` function. | | | | |

| **#5** | User password allows for length 33. | Type | **White box** | Severity | **3** |
|---|---|---|---|---|---|
| Output before seeding | Server will check the length of the password, and ensure the length to be no less than 6 and no more than 32. | | | | |
| Output after seeding | Server will allow for a string of length 33 as password. | | | | |
| Suggested Correction | In `class.Core.php`, in `isValidPassword` function, change the number 33 to 32. | | | | |

| **#6** | Server will reveal debug information. | Type | **White box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | Server will hide all errors, failures, and other exceptions from client, but record such problems in server log. | | | | |
| Output after seeding | Server will print all errors directly to stderr, which is actually the browser or HTTP requester. For a production branch this is dangerous and will be a serious mistake. | | | | |
| Suggested Correction | Turn off debug mode in `config.inc.php`. | | | | |

| **#7** | When database server can't be connected, the user credentials for the database will be revealed. | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | If such connection failure happens, error will be written to `php_error.log`, and the execution is terminated, leaving the page blank. | | | | |
| Output after seeding | The connection failure will be printed directly on the page, including the database host and port number, username, and password, and the database name. | | | | |
| Suggested Correction | In function `connect()` of `class.Database.php`, change the code when the database object is not instantiated to `die()` directly, instead of printing the details. | | | | |

| **#8** | Server always returns a list of users who are in the default range of distance to the requester. | Type | **Black box** | Severity | **2** |
|---|---|---|---|---|---|
| Output before seeding | Client will send a parameter indicating the desired range, and server searches in the database for those who are in such range. | | | | |
| Output after seeding | The desired range parameter is ignored by the server; server will always search for users within the default value of range. | | | | |
| Suggested Correction | Use the desired range parameter given by the requester to overwrite the value of the range variable in `action.php`. | | | | |

| **#9** | Server does not distinguish user email and username parameters. | Type | **Black box** | Severity | **3** |
|---|---|---|---|---|---|
| Output before seeding | Server strictly distinguish usernames from user emails because they have different format requirements. | | | | |
| Output after seeding | If one searches for an email address but gives a username in that field, server will return the user whose username matches that value. | | | | |
| Suggested Correction | Change all the combined checks in `class.User.php`. | | | | |

## Client-side Defects

| **#1** | The passwords sent by client are not encrypted. | Type | **Black** | Severity | **1** |
|---|---|---|---|---|---|

| | |
|---|---|
| Output before seeding | The password was encrypted before sending so that even if a hacker gets the HTTP packets, the data is mostly useless to him. |
| Output after seeding | In HTTP POST data of events like login or registration, the original password will show up. E.g., if the `password` is "`123456`", there will be a field like "`password=123456`". |
| Suggested Correction | In `APIHandler` class, add `md5` encryption to the related functions like `login` or `register`. |

| **#2** | The ListView in home activity cannot be displayed correctly | Type | **Black** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | The ListView should be displaying the user icon and several user info | | | | |
| Output after seeding | The ListView display nothing | | | | |
| Suggested Correction | Modifying the home activity class, adding adapter | | | | |

| #3 | The server shutdown caused client response string to null and handle the string incorrectly will crash the client | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | When the server shutdown, the null value of the response string is correctly handled and a login error message showed on the screen | | | | |
| Output after seeding | When the server shutdown, the null value caused NullPointerException and crash the client app. | | | | |
| Suggested Correction | Add the correct null value check before accessing the response string | | | | |

| #4 | The network availability is not checked by the client, if a network is not available, it will cause the crash of the app. | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output | If the network is not available, the client will prompt the message and | | | | |

| | |
|---|---|
| before seeding | login or register cannot be executed. |
| Output after seeding | If the network is not available, the client will still try to access the network and cause the client crashed. |
| Suggested Correction | Use the network availability method to check the network before trying to access the network. |

| | | | | | |
|---|---|---|---|---|---|
| #5 | In ListView layout, the TextView displaying details of hobbies may overlap with the User Image if the list of hobbies are too long. | Type | **Black box** | Severity | **3** |
| Output before seeding | The hobbies should not overlap with the user Icon | | | | |
| Output after seeding | The hobbies overlap with the user Icon if it is too long | | | | |
| Suggested Correction | Modify the position of TextView of hobbies | | | | |

| | | | | | |
|---|---|---|---|---|---|
| #6 | The line overlapped with the top of ListView. | Type | **Black box** | Severity | **3** |
| Output before seeding | Overlapping should not be happening | | | | |
| Output after seeding | Overlapping happened | | | | |
| Suggested Correction | "android:layout_below="@+id/textViewLine" | | | | |

| | | | | | |
|---|---|---|---|---|---|
| #7 | The button will disappeared if the username is too long | Type | **Black box** | Severity | **2** |
| Output before seeding | The button should not disappear if the username is too long. It will overlap with the username. | | | | |
| Output after seeding | The button is being pushed out of the boundary due to the wrong align. | | | | |
| Suggested | Modified android:layout_marginRight=”21dp” | | | | |

| | |
|---|---|
| Correction | |

| | | | | | |
|---|---|---|---|---|---|
| #8 | The ListView cannot display the last item correctly | Type | **Black box** | Severity | **1** |
| Output before seeding | The ListView should display the same contents for every items. | | | | |
| Output after seeding | Some contents is being cut-off due to the size of ListView | | | | |
| Suggested Correction | Modify the size(width, height) of the ListView | | | | |

| | | | | | |
|---|---|---|---|---|---|
| #9 | The orientation of the signupActivity does not change to the horizontal when the orientation of the phone change to the horizontal | Type | **Black box** | Severity | **3** |
| Output before seeding | The signupActivity changed to the horizontal when the phone changed to the horizontal. | | | | |
| Output after seeding | The signupActivity does not change the direction when the phone changed to the horizontal. | | | | |
| Suggested Correction | Add the code to handle the orientation changes. | | | | |

| | | | | | |
|---|---|---|---|---|---|
| #10 | In ListView layout, the TextView displaying details of hobbies may overlap with the User Image if the list of hobbies are too long. | Type | **Black box/ White board** | Severity | **2** |
| Output before seeding | They should not be overlapping and should be displayed correctly | | | | |
| Output after seeding | They are overlapped with each other | | | | |
| Suggested Correction | Modified by using align. | | | | |

| | | | | | |
|---|---|---|---|---|---|
| #11 | In the Sign Up Upload Fragment, when no image on the phone, the | Type | **Black box** | Severity | **3** |

| | | | | | |
|---|---|---|---|---|---|
| | app does not have error message shows up | | | | |
| Output before seeding | If no image on the phone, a message box should show up and tell the user to upload some photos to the phone | | | | |
| Output after seeding | If no image on the phone, nothing will show up and signup activity cannot be finished. | | | | |
| Suggested Correction | Add a method to show up a message box in the signup fragment when there is no image on the phone. | | | | |

| #12 | The home_activity is not displaying the correct user information | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | The home_activity should be able to display the username and hobbies. | | | | |
| Output after seeding | | | | | |
| Suggested Correction | | | | | |

| #13 | The signup fragment does not check the validity of the email address | Type | **Black box** | Severity | **2** |
|---|---|---|---|---|---|
| Output before seeding | If the email is in the incorrect form, an error message will show up | | | | |
| Output after seeding | If the email is in the incorrect form, no error message will show up and the client will send the incorrect email to the server | | | | |
| Suggested Correction | | | | | |

| #14 | | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | | | | | |
| Output after seeding | | | | | |
| Suggested | | | | | |

| Correction | |
|---|---|

| #15 | | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | | | | | |
| Output after seeding | | | | | |
| Suggested Correction | | | | | |

| #16 | | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | | | | | |
| Output after seeding | | | | | |
| Suggested Correction | | | | | |

| #17 | | Type | **Black box** | Severity | **1** |
|---|---|---|---|---|---|
| Output before seeding | | | | | |
| Output after seeding | | | | | |
| Suggested Correction | | | | | |