"Bonjour"

Sprint II
Defect Logs

Last Revised: February 21, 2014

Team 7 Members:

Xiangyu Bu

Rishabh Mittal

Yik Fei Wong

Yudong Yang

Overview

Purpose

This document is to record all the defects found during sprint 2 inspections.

Design Inspection Defects

Inspection s2.d1

Summary

- Product: Bonjour / Product Backlog
- Date: Feb 16, 2014
- Author: Bonjour Team
- Moderator: Xiangyu Bu
- Inspector: Yudong Yang, Yik Fei Wong, Rishabh Mittal
- Recorder: Yudong Yang

Defects

#	Severity	Description	Solution
1	1	The design of password retrieval mechanism has a flaw that if an attacker sends to the server a password retrieval request of an arbitrary user, a new password will be sent to this user's account; although the password is not leaked, it annoys the actual user.	Designed a new mechanism, written as the note #1 of this inspection log (see below).
2	3	It would be better if the user can get an email confirmation when his / her password changes.	Added.
3	2	The user profile set needs a summary.	Basic fields: • Username, email • User avatar • User hobby Extended fields: • Birthday • MaximumRange • Phone number

Notes

1. The new mechanism for resetting a forgotten password:

   - Client asks for the email of which to reset password, and sends it to the server
   - Server returns the security question set by the user to the client
   - Client shows the question on UI, and asks the user to answer it and submit
   - Server verifies the question, randomly generates a new password, and sends it to the user's email address.
   - However, if the security question is not set, the system will return an error number indicating the client to ask if the user wants to generate a new password and send it to the user's email address, and does this if confirmed.

2. Thus far the user identity part should be well-defined.

Inspection s2.d2

Summary

- Product: Bonjour / Product Backlog
- Date: Feb 18, 2014
- Author: Bonjour Team
- Moderator: Yudong Yang
- Inspector: Yudong Yang, Yik Fei Wong, Xiangyu Bu
- Recorder: Xiangyu Bu

Defects

#	Severity	Description	Solution
1	1	The specifications for user avatar is not defined yet.	The following specifications are added for user avatar image: • Must be one of the following format: JPEG, GIF, PNG (in regular expression, `/\.(gif|jpe?g|png)$/i`) • File size must be smaller than 200KiB, and greater than 0B • Only the last uploaded image is stored

2	3	User cannot crop avatar picture.	Will add crop feature to both server side and client side.
3	1	The user hobby is not defined in great detail. It is time to make this concrete.	The user hobby include: • Reading • Writing • Programming • Dance • Music • Games • Drama • Sports • Collection
4	1	The data type for location is not defined yet. It is time to make this concrete.	Based on the Location object of Android sdk source, the distance can be calculated as a function of mLatitude, mLongitude, and mAltitude. It also includes a time field named mTime. The server will also use this Location class as the data type for location.
5	1	We need to define clearly the matching rules at this stage.	The rules are as follow: • User will specify the range within which the app will match other users • Server will generate the list of users who is within the range and have hobby overlap with the user • Server will return the list Client is responsible for rendering and sorting the list to customize the user's needs
6	2	The user agreement (terms of service) is not written yet.	Added to list, and links prepared. Will write the actual content when the time comes.

Notes

Inspection s2.d3

Summary

- Product: Bonjour / Test Plan, Incremental Test Log
- Dates: Feb 19 and Feb 20, 2014
- Author: Bonjour Team

- Moderator: Yik Fei Wong
- Inspector: Xiangyu Bu, Yik Fei Wong, Yudong Yang, Rishabh Mittal
- Recorder: Yudong Yang

Defects

#	Severity	Description	Solution
1	3	Allowing the email string to be as long as 255 chars is unnecessary and might break the database.	Restrict the length to 127 chars.
2	3	Having used separate LoginException and RegisterException, and adding ResetPasswordException and maybe more in the future, it is hard to manage and test them without trouble.	Created an abstract class named `UserException`, which models the general exception event in `User` class, and let all other Exceptions from User class extend it. This extra abstraction layer will reduce the test work.
3	2	It would be better if there is a web UI for the faker. This way we can use Selenium IDE to automize some testing work.	Made a simple UI which conforms to the server API. It can be used and will only be used for testing.

Code Inspection Defects

Inspection s2.c1

Summary

- Product: Bonjour / Client Side
- Date: Feb 17, 2014
- Moderator: Yudong Yang
- Inspector: Yudong Yang, Yik Fei Wong
- Recorder: Xiangyu Bu

Defects

#	Severity	Description	Solution
1	1	A null context passed into SQLHandler, then a database cannot be initialized	Pass the correct context into the structure of SQLHandler and check if the context is null or not at the SQLHandler, throw the NullPointerException when it is null

2	3	SQLHandler does not check the validity of the user data before inserting into the local database	Check the validity of data before inserting into the database
3	2	SQLHandler needs to handle null values in the Bundle object, If a value is not wrapped in the Bundle object, it will cause a crash because of the NullPointerException	Check the null values in the Bundle object and Bundle object itself to make sure the function will not access a null value and it can return a false boolean value back to the caller function.
4	1	A typo in the SQLHandler caused the function tries to insert a non-exist attribute into the database and caused an exception	Fix the typo to ensure the correct attribute is used in the function
5	3	The client does not check user age before sending registration requests.	Added.

Notes

Inspection s2.c2

Summary

- Product: Bonjour / Server Side
- Date: Feb 19, 2014
- Moderator: Xiangyu Bu
- Inspector: Yudong Yang, Xiangyu Bu
- Recorder: Yik Fei Wong

Defects

#	Severity	Description	Solution
1	1	File upload class may be subject to attack by uploading non-image files.	Wrote code to check the extension name of the uploaded files.
2	1	File upload class did not have code to crop the large image.	Wrote code to initiate gd library to process the image.
3	2	File upload class cannot recognize aborted http file	Wrote code to check the reported file size and actually

		upload requests.	received file size.
4	3	File upload class stores the file with its original name.	Wrote the code to rename the file before saving to disk.
5	3	File upload class did not have a limit for file size.	Added code to check file size, thus not relying on PHP configuration anymore.
6	3	There is memory leak in the customized exceptions like `LoginException` and `RegisterException`.	Fixed.
7	2	The password change method `User::changePassword` does not verify the format of the username and passwords fed.	Standardized the verification methods and added the related verification code.
8	2	The server does not check the user age when handling registration requests, either.	Added.
9	3	The `LoginException`, `RegisterException`, `ResetPasswordException` are of same structure, but are handled in different methods. This makes it hard to keep track of code.	Wrote a new method to handle the type `UserException` directly. With polymorphism the handler of those exceptions can be unified.

Notes

Unit Testing Defects

Inspection s2.ut1

Summary

- Product: Bonjour / Server Side
- Date: Feb 19, 2014
- Author: Bonjour Team
- Moderator: Xiangyu Bu
- Inspector: Yudong Yang, Xiangyu Bu
- Recorder: Yudong Yang

Defects

#	Severity	Description	Solution
1	3	When using faker to upload files, the file upload handler class accepts non-image files.	Added restrictions to file extension names.
2	3	File upload class cannot disregard aborted HTTP file upload requests.	Added the rule that if the file size does not match the data received, abort handling the request.
3	2	When uploading a valid image file, the stored file was not resized.	The server did not activate gd library of PHP. Activated it.
4	1	When uploading files anonymously, the file upload handler class accepts the file in the temporary directory.	Now the dispatcher will check if the user has logged in with a valid access_token before instantiating the handler. If the uploader is not a valid user the request will not be accepted.
5	2	File upload class cannot distinguish the operation of uploading a non-file object (e.g., a folder, faker text that does not conform to HTTP FILE specs).	Added a checker for this particular situation.
6	1	When uploading a php code file renamed to .png extension name, file upload class accepts the request the does stores it to disk, but it cannot be executed.	Added a checker for the MIME type, and on server side removed execution permission of uploaded files and the parent directory.
7	2	When handling the request of register an user whose username reaches the maximum length, the access token generated becomes 0, which is incorrect.	The built-in hash functions has implicit length restrictions, and violating the rule gets a null return value. Fixed it by adding a pre-hashing function.

Notes

Not being able to handle aborted HTTP requests is not a big issue in server, but it has to be thought of carefully now that the project builds a mobile app.

Inspection s2.ut2

Summary

- Product: Bonjour / Client Side
- Date: Feb 20, 2014
- Author: Bonjour Team
- Moderator: Yudong Yang
- Inspector: Yudong Yang, Yik Fei Wong
- Recorder: Xiangyu Bu

Defects

#	Severity	Description	Solution
1	1	Closed the database object accidently caused the SQLHandler can only handle one transaction	Open the database object initially and keep the database opened until transactions finished
2	3	The orientation of the Login activity cannot be changed	Add the orientation property to the UI xml to allow the login activity can be changed to horizontal
3	2	The back button of each activitiy is not associated with the previous activity	Combine the back button with correct previous activity

Notes