"Bonjour"
Project Backlog

Last Revised: March 08, 2014

Team 7 Members:

Xiangyu Bu

Rishabh Mittal

Yik Fei Wong

Yudong Yang

Problem Statement

As technology advances, people have been spending increasingly more time with the virtual world and to some extent, neglected to meet people who share similar interests near them. The "Bonjour" app, which we are to develop, aims to help people notice whether they have met anyone nearby who shares common interests with them. It also provides a chance for people to develop potential friendship with others.

Background Information

As is said, wth the prevalence of mobile technology the virtual world is encroaching our real social life. Many software developers have noticed this, and tried to develop software to strengthen people's physical social connection. To make Bonjour stand out, we must beware similar apps in the market.

Applications like WeChat and WhatsApp have provided similar set of functionalities as Bonjour. They allow users to search and chat with nearby users, but do not provide any matching according to users' interests. But Bonjour will match users' common interest and display a list of potential friends.

Also, in those two apps, finding friends is mostly an add-on of the app, and users need to click the search button to perform searching. By contrast, in Bonjour, matching is the center, and is done automatically. Whenever there is someone who share's similar interests nearby, the application automatically matches the two users and send them a notification.

During later research of the project, we found that there is an app called Jack'd, which is similar to Bonjour in terms of functionality set. However, we have totally different target users. The target users of Jack'd are males who are willing to find male partners. For Bonjour, we do not limit our users, and does not encourage Jack'd's user pattern. Everyone who wishes to find new friends or project partners or any personal relationship can use Bonjour to their advantage.

Environment and System Model

Environment
- Client: Android 4.0.3 or newer, written in Java with Android SDK
- Server: PHP5 and MySQL5.6 on Apache 2 httpd

System Model: C/S architecture
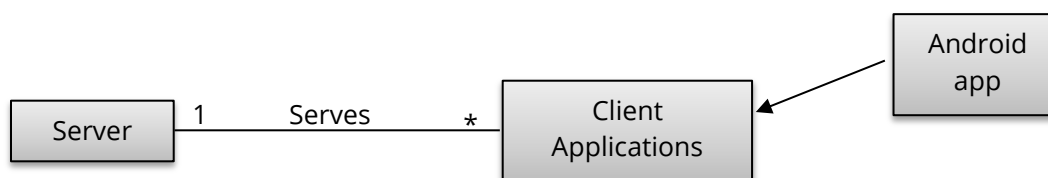
Requirements

The requirements are categorized by the functional components of our application. Inside each major functional component, we have these types of requirements: functional requirements, and non-functional requirements, of which the latter includes software quality requirements, platform requirements, and process requirements.

User Identity

"User identity" is the set of functionalities the user identifies oneself with the application.

Basic properties:

a. User email is a string of the form
   `[-0-9a-zA-Z.+_]+@[-0-9a-zA-Z.+_]+\.[a-zA-Z]{2,4}`
b. User password is a string of length 6 to 32, allowing special characters.
c. User nickname is a string of length 4 to 20 of the form
   `[0-9a-zA-Z.+_]{4, 20}`
d. User age must be within the range of 13 to 100.

Basic behavior:

1. user registration
2. user log in
3. password reset
4. profile update

The following is the list of user stories.

User.R01.Android				Not started	
As a user, I want to register my own account.					
Category	Functional	Priority	High	Hours	**6**
User story	Related Use Cases: User.L01.Android Steps: 	Actor actions	System responses
---	---
1. Click the register button and agree with terms and conditions	2. The register activity shows up
3. Enter the username and password and click "Next"	4. Move to the create profile UI
5. Enter the basic user information and click "Next"	6. Move to the photo upload UI
	8. The photo shows on the screen
	10. Pop up a register result					

	7. Choose a photo on the phone	message and return back to the login page
	9. Click the confirm register button	
Comment		

User.L01.Android   In Progress

As a user, I want to log in with my account and password.

Category	Functional		Priority	Normal	Hours	**4**

User story	Related Use Cases:
	Steps:

Actor actions	System responses
1. Run the app	2. Login page shows up
3. Enter the username and the password, click "Log in" button	4. Check the validity of username and password, log into the account or shows a failure message

Comment	

User.RP.Android   Not started

As a user, I want to reset my password when I want to.

Category	Functional		Priority	Low	Hours	**5**

User story	Related Use Cases:
	Steps:

Actor actions	System responses
1. Run the app	2. Home Page shows up
3. Click the "My Profile" button	
4. Click "Reset Password" and confirm	5. Send a link to user's email for resetting password
6. Open email and reset password by clicking on the link	7. Replace the old password with the new one

	and setting new password			
Comment				

User.RP2.Android				Not started	
As a user, I want to reset my password when I could not log in.					
Category	Functional	Priority	Low	Hours	**3**
User story	Related Use Cases: Steps:<table><tr><td>Actor actions</td><td>System responses</td></tr><tr><td>1. Run the app 3. Click the "Reset my password" link 5. Type email address and confirm 8. Answer the question</td><td>2. Home Page shows up 4. System asks for the email address 6. return a security question set by the user 7. display the question 9. Verify the answer, if correct, generate random password and send to user's email.</td></tr></table>					
Comment					

User.RP3.Android				Not started	
As a user, I want to reset my password when I could not log in, but I did not have my security question set.					
Category	Functional	Priority	Low	Hours	**3**
User story	Related Use Cases: Steps:<table><tr><td>Actor actions</td><td>System responses</td></tr><tr><td>1. Run the app 3. Click the "Reset my password"</td><td>2. Home Page shows up</td></tr></table>					

	link 5. Type email address and confirm 7. Choose yes	4. System asks for the email address 6. return an error to ask if the user want to generate a new password and send to user's email address 8. Generate random password and send to user's email.
Comment		

User.PU.Android					Not started	
As a user, I want to view my profile.						
Category	Functional		Priority	Normal	Hours	**7**
User story	Related Use Cases: 1. (User.RP.Android) resetting password Steps: <table><tr><td>Actor actions</td><td>System responses</td></tr><tr><td>1. Run the app 3. Click the "My Profile" button</td><td>2. Home page shows up 4. Provides user with the daily matches and required information</td></tr></table>					
Comment						

Matching

The android app will collect the location information, and upload it to the server for matching purposes.

The basic behavior includes:

- Notification of location gathering
- Location gathering
- Showing a list of people nearby
- View the profile of those who are close to the user
- Matching between 2 users.

Match.Lg.Android   Not started

As a user, I want to upload my location information to the server and see the result.

Category	Functional	Priority	Normal	Hours	**4**

User story	Related Use Cases: Steps:		
	Actor actions		**System responses**
	0. No specific actions needed		

Comment	(XB) We should provide a button to manually trigger location update.

Match.PV.Android   Not started

As a user, I want to see those who are close to me.

Category	Functional	Priority	Normal	Hours	**4**

User story	Related Use Cases: Steps:		
	Actor actions		**System responses**
	1. Run the app		2. Homepage shows up 3. The system updates the current location and syncs with the server 4. Refresh the list of nearby users

Comment	

Match.PV2.Android   Not started

As a user, I want to view the profiles of those who are close to me.

Category	Functional	Priority	Normal	Hours	**8**

User story	Related Use Cases: 1. (User.PU.Android) View and update my profile.

	Steps:	
	Actor actions	System responses
	1. Open the app 2. Notice the names of the people who share similar interests. 3. Click on a particular user avatar to view profile 5. Check out the profile and initiate a chat with the match user	4. Extract matched user profile from database and send information
Comment	The user can also view the profile by clicking the avatar of the user list in the homepage.	

Match.M1.Android				Not started	
As a user, I want to match people who shared similar interest with me.					
Category	Functional	Priority	High	Hours	**5**
User story	Related Use Cases: Steps:				
	Actor actions		System responses		
	1. Open the app 6. Display the result		2. Application send GPS information to the server 3. Server search for nearby users 4. Server compare their interest 5. At most 3 best matches are send to the client		
Comment					

Other General, Non-functional Requirements
The following non-functional requirements apply to all the components of the project and must be pondered over at each step.

1. **Security**. Information that needs to be restored will be encrypted by RSA algorithm, while non-restorable information will be encrypted by MD5 algorithm. The request of information will need clearly defined permission level to be performed.
2. **Power Saving**. If not designed will, the app may become a battery killer.
3. **Stress**. Carefully design the algorithm so that the matching and listing operations will not cause the server and client to crash.
4. **Accuracy**. Be sure to enhance the accuracy of location information with the help of Wi-Fi and Bluetooth. Accurately reporting location is critical to the app.