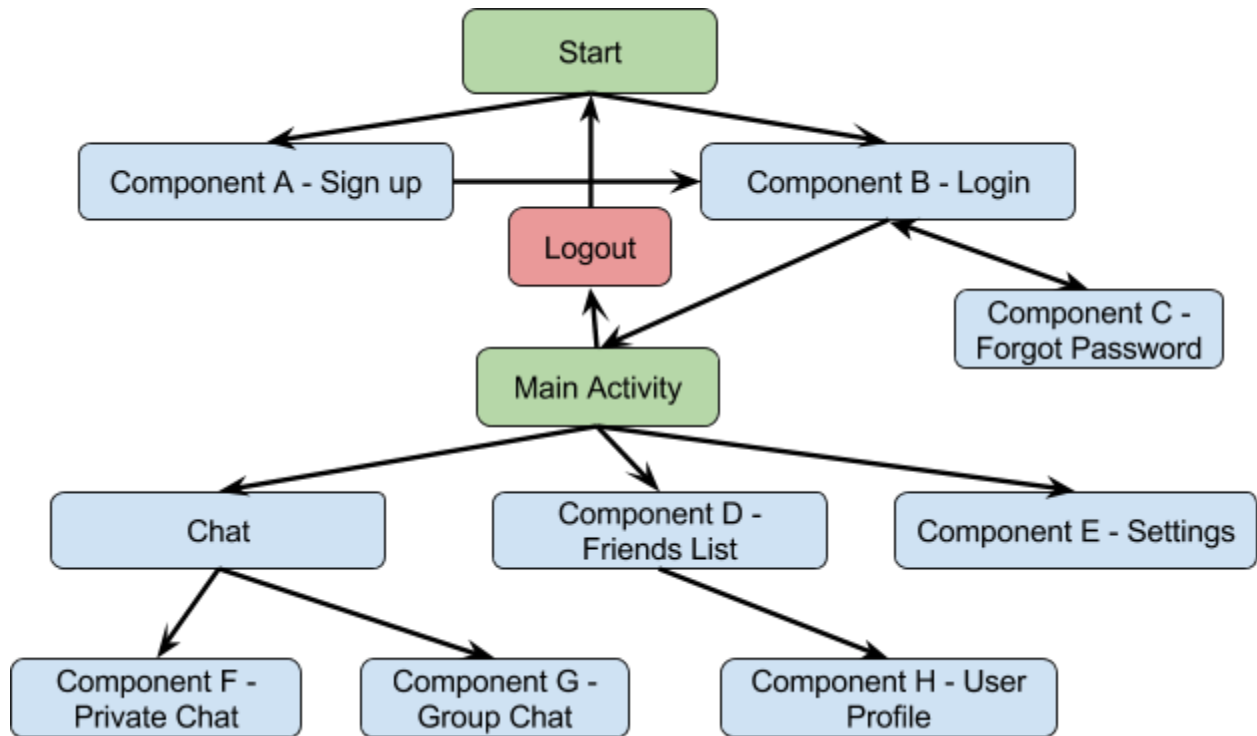


## *Team 17 - WeGroup*

# Incremental Testing & Regression Testing

Hanqi Du, Ao Liu, Xiaoyang Lin, Wanqi Jiang, Qinxin Shi, Zhixuan Yu

### Classification of components



- Start
  1. Input: None
  2. Output: None
  3. Parent dependency: None
  4. Child dependency: Sign up and Login
- Component A - Sign up:
  1. Input: email, user name, password
  2. Output: status of sign up
  3. Parent dependency: Start
  4. Child dependency: Login

- Component B - Login:
  1. Input: email, password
  2. Output: status of login
  3. Parent dependency: Start, Sign up, Forgot Password
  4. Child dependency: Forgot Password, Main Activity
- Component C - Forgot Password:
  1. Input: Email address
  2. Output: None
  3. Parent dependency: Login
  4. Child dependency: Login
- Component D - Friends List:
  1. Input: userId
  2. Output: Id of this user's friends
  3. Parent dependency: Main activity
  4. Child dependency: User profile
- Component E - Settings:
  1. Input: Short bio, birthday, gender
  2. Output: Display the updated user profile
  3. Parent dependency: Main Activity
  4. Child dependency: None
- Component F - Private Chat:
  1. Input: Chat message, user id, timestamp
  2. Output: message text, from user, timestamp
  3. Parent dependency: Chat
  4. Child dependency: None
- Component G - Group Chat:
  1. Input: Group chat message, group chat id, timestamp
  2. Output: message text, from user, timestamp
  3. Parent dependency: Chat
  4. Child dependency: None
- Component H - User Profile:
  1. Input: None
  2. Output: user name, short bio, gender, birthday

3. Parent dependency: Friend list
4. Child dependency: None

### **Top-down Incremental Testing**

In this project, we followed top-down incremental testing. We tested the app from the Start to the more in-depth components of the application. This method of testing allowed us to discover defects of the major parts of the program and make sure that the structure works fine in the early stage of implementation, which made the testing process for smaller components easier.

### **Incremental & Regression Testing**

Module	Component A - Sign up		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	User should not be able to sign up with an email address that has already been signed up	2	Every time there is a register request, server should check the database to see if the email already exists. If so, send an error message back to the client.
2	User should see a warning when the the conformation password is different from the first one	2	The front end application checks whether two texts are the same before sending it to server, and if they are not, the system should pop up a warning message
Regression Testing			
Defect #	Description	Severity	How to correct
1	Fixing duplicate email address error, normal signing up sometimes behaves abnormally	2	The client clears the email address every time after sending it to the server

Module	Component B - Login		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Log in successfully with empty email and password	2	Add check statement to check the length of the user name and password in the log in function.
Regression Testing			

Defect #	Description	Severity	How to correct
1	After fixing logging in successfully with empty email and password, user cannot login with the email and password they signed up.	1	Add check statement in the sign up function to make sure users use the correct length of email addresses and passwords.

Module	Component C - Forgot Password		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	The application should not crash when the email entered to retrieve password is not exist in the database.	1	When check the email entered by user in the database, if the email entered is not found, return invalid value to the server.
Regression Testing			
Defect #	Description	Severity	How to correct
1	Fixing invalid email address, but the server cannot send email to user.	3	Server send an error message to client by displaying a message shows the email address has not been signed up and asked user to choose either re-enter the email address or sign up with this email address.

Module	Component D - Friends List		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	App should not crash when the client fails to receive or parse certain information (e.g. user name) and cannot display	1	Add optional value check. When a value is not provided by the server, “unknown” should be displayed
2	The list should not contain a recently deleted friends	2	Locally remove the deleted friend entity after calling the server to remove remotely
3	The list should contain newly added friends	2	After sending the request to server, add the added friend in the list locally instead of calling the server for the whole list

Regression Testing			
Defect #	Description	Severity	How to correct
2	Removing the “deleted friends” locally sometimes causes inconsistency with the backend	1	Only remove the user after receiving the success message from server
3	Adding friends locally causes abnormal behaviors on Component H	2	Add a global variable and set up some protocols to globalized user information

Module	Component F - private chat		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	The application should not allow user send an empty message or a message exceeds 140 words.	2	Add conditions to handle the case after clicking the send button. If the length of message is not in valid range, print the “the message cannot be empty” or “the message is too long” respectively on the display.
Regression Testing			
Defect #	Description	Severity	How to correct
1	When user send message too rapidly in very short period, the receiver will received the message in wrong order.	1	Use a correct protocol to fix the problem.

Module	Component G - Group Chat		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	When invite a friend to a group, the UI display the friend was in the group but the user cannot receive message from the group.	2	Everytime, when a user invited a friend into the group, the server need to update the friend's group field in the database before the UI display.
Regression Testing			
Defect #	Description	Severity	How to correct

--	--	--	--

Module	Component H - User profile		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	When an user changed his profile, his friend cannot view the newest version.	2	Every time a user view other user's profile,client should request the newest profile data from server.
2	App should not crash when the user does not have a short bio (unwrapping nil value)	1	Check for nil values on the client side and make sure not to unwrap nil values. Display an empty string if the short bio is nil
Regression Testing			
Defect #	Description	Severity	How to correct
1	Fixing this problem causes Component D to display incorrect information	1	Added segue to update the list value in the friends list

Module	Component E - Settings		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	User should be able to set short bio. Leave and coming back to the settings page, the bio should be set to the new one	2	Add check statement to check whether the server send the new short bio successfully. If not, A popup should appear to inform the user.
Regression Testing			
Defect #	Description	Severity	How to correct
1	Fixing this error sometimes causes the Component H behaves abnormally	2	The profile page should check for updates every time it is opened up