Ranged Battle Chat

Wesley Bales, Leonardo Guajardo,
Brad Mickow, Thomas Wolf

UT DALLAS

Demonstration

- Login Attempt for non-user
- Account Creation
- Login
- Send Message

Git Repository

- Show Git commits
- Pull requests
- Commit messages

Team Organization and Design

- Met weekly after class for quick touch and go meetings
- Communicated through GroupMe
- Used pair programming and often worked as a group on tasks

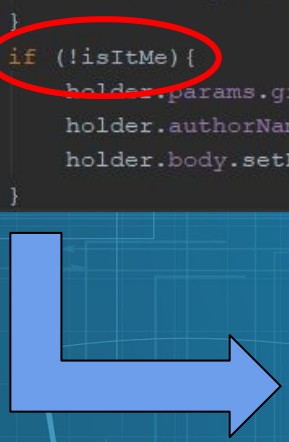
Coding Styles

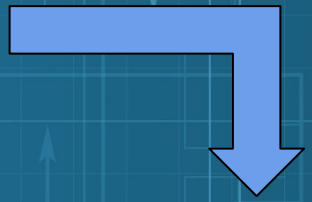
- Primary coding convention was camelCase
- JavaDoc was used in Android Studio java files
- We mostly used comments to describe classes and methods, and did not write comments for all lines.

```java
/**This method decides the appearance of the message on the screen. Sent messages are in a green box and
 * received messages are in a blue box.
 * @param chatFromSender - Boolean value to decide the holder is the sender or receiver of a message
 * @param holder - ViewHolder object that contains a TextView for the author's name and the contents of a
 *               message.
 */
private void setChatRowAppearance(boolean chatFromSender, ViewHolder holder) {

    if (chatFromSender) {
        holder.params.gravity = Gravity.END;
        holder.authorName.setTextColor(Color.BLACK);
        holder.body.setBackgroundResource(com.rangedbattle.firebasechat.R.drawable.bubble2);
    } else {
        holder.params.gravity = Gravity.START;
        holder.authorName.setTextColor(Color.BLACK);
        holder.body.setBackgroundResource(com.rangedbattle.firebasechat.R.drawable.bubble1);
    }

    holder.authorName.setLayoutParams(holder.params);
    holder.body.setLayoutParams(holder.params);

}
```

Refactoring

```
private void setChatRowAppearance(boolean isItMe, ViewHolder holder) {

    if (isItMe) {
        holder.params.gravity = Gravity.END;
        holder.authorName.setTextColor(Color.BLACK);
        holder.body.setBackgroundResource(com.rangedbattle.firebasechat.R.drawable.bubble2);
    }
    if (!isItMe){
        holder.params.gravity = Gravity.START;
        holder.authorName.setTextColor(Color.BLACK);
        holder.body.setBackgroundResource(com.rangedbattle.firebasechat.R.drawable.bubble1);
    }
```





```
private void setChatRowAppearance(boolean chatFromSender, ViewHolder holder) {

    if (chatFromSender) {
        holder.params.gravity = Gravity.END;
        holder.authorName.setTextColor(Color.BLACK);
        holder.body.setBackgroundResource(com.rangedbattle.firebasechat.R.drawable.bubble2);
    } else {
        holder.params.gravity = Gravity.START;
        holder.authorName.setTextColor(Color.BLACK);
        holder.body.setBackgroundResource(com.rangedbattle.firebasechat.R.drawable.bubble1);
    }
```

Testing

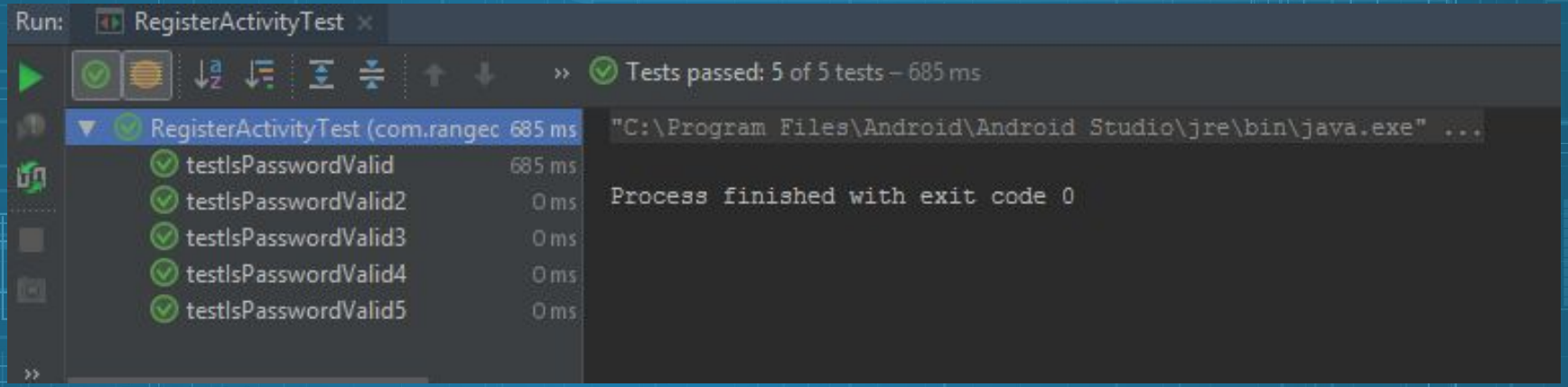
```
@VisibleForTesting(otherwise = VisibleForTesting.PRIVATE)  boolean isPasswordValid(String password, String confirmPassword) {
    //TODO: Add own logic to check for a valid password

    return confirmPassword.equals(password) && password.length() > 4 && (password.contains("!")||password.contains("@")||
            password.contains("#")||password.contains("$")||password.contains("%")||password.contains("^")||password.contains("&")
            ||password.contains("*")||password.contains("(")||password.contains(")"));
}
```

This is the method that was chosen for the unit testing phase RegisterActivity.isPasswordValid it must be at least four Characters long and contain at least one symbol .

Testing

```
@Test
public void testIsPasswordValid() {
    assertEquals( expected: true, tester.isPasswordValid( password: "Test!", confirmPassword: "Test!"));
}
@Test
public void testIsPasswordValid2(){
    input = "t@est";
    assertEquals( expected: true,tester.isPasswordValid(input,input));
}
@Test
public void testIsPasswordValid3(){
    input= "tes#$";
    assertEquals( expected: true,tester.isPasswordValid(input,input));
}
@Test
public void testIsPasswordValid4(){
    input= "test";
    assertEquals( expected: false,tester.isPasswordValid(input,input));
}
@Test
public void testIsPasswordValid5(){
    input= "$^&";
    assertEquals( expected: false,tester.isPasswordValid(input,input));
}
```

Testing



All five test passed, three expected True, two expected False.

Q&A

Contact

Wesley Bales - wrb160330@utdallas.edu
Leo Guajardo -
Brad Mickow - bam171230@utdallas.edu
Thomas Wolf - trw160130@utdallas.edu