Assignment: Portfolio of Evidence Part 2
Due date: 2025/04/26
Name: Blessing Jafari Mbalaka
Student Number: ST10466840
Assignment Instructions: Java Messaging Class

SECTION A-STUDENT DETAILS

Student Number: ST10466840
Name(s): Blessing Jafari
Surname: Mbalaka
Assessment: POE Part 2 [Message class]

Links to Github and Youtube Video:

- **Private repo Github link also here:** https://github.com/VCPTA/bca1-prog5121-part-1-submission-ST10466840/pull/new/feature/Login-message-class
- **Youtube Video of Unit Program code explanation/Walkthrough and Unit tests:**
- **Unit Test video:** https://youtu.be/3Cku_IrtTkA
- **Code Running Video:** https://youtu.be/LBmqUpDA08ghttps://youtu.be/LBmqUpDA08g

---

Section B-Full Source Code

```
package com.mycompany.messagingapp;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;

public class Messagingapp_ {
    public static void main(String[] args) {
        // 1) perform login flow
        Login.runLogin();

        // 2) welcome
        JOptionPane.showMessageDialog(
            null,
            "Welcome to QuickChat.",
            "Welcome",
            JOptionPane.INFORMATION_MESSAGE
        );

        MessageStore.init();
        boolean running = true;

        while (running) {
            String[] options = {"Send Messages", "Show Recently Sent Messages", "Quit"};
            int choice = JOptionPane.showOptionDialog(
                null,
                "Choose an option:",
                "Main Menu",
                JOptionPane.DEFAULT_OPTION,
                JOptionPane.QUESTION_MESSAGE,
                null,
                options,
                options[0]
            );

            switch (choice) {
```

```
            case 0: // Send Messages
                sendMessagesFlow();
                break;
            case 1: // Show recently sent
                JOptionPane.showMessageDialog(
                    null,
                    "Coming Soon.",
                    "Feature In Development",
                    JOptionPane.INFORMATION_MESSAGE
                );
                break;
            default: // Quit or closed
                running = false;
        }
    }

    // on exit: export JSON
    MessageStore.exportJson();
    System.exit(0);
  }

  private static void sendMessagesFlow() {
    String input = JOptionPane.showInputDialog(
        null,
        "How many messages will you send?",
        "Send Messages",
        JOptionPane.QUESTION_MESSAGE
    );
    int max;
    try {
        max = Integer.parseInt(input.trim());
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null, "Invalid number.", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    for (int i = 1; i <= max; i++) {
        String title = "Message " + i + " of " + max;
        String recipient = JOptionPane.showInputDialog(null, "Recipient (+…):", title,
JOptionPane.QUESTION_MESSAGE);
        String content   = JOptionPane.showInputDialog(null, "Message (≤250 chars):", title,
JOptionPane.QUESTION_MESSAGE);

        Message m = new Message(Message.generateId(), i, recipient, content);

        // validations
        if (!m.checkMessageID()) {
            JOptionPane.showMessageDialog(null, "Message ID invalid (max 10 characters).", "Validation Error",
JOptionPane.ERROR_MESSAGE);
            i--; continue;
        }
        if (!m.checkRecipientCell()) {
            JOptionPane.showMessageDialog(null, "Recipient invalid (max 10 chars, must start with +).",
"Validation Error", JOptionPane.ERROR_MESSAGE);
            i--; continue;
        }

        String[] sendOpts = {"Send", "Discard", "Store"};
```

```
            int sendChoice = JOptionPane.showOptionDialog(
                null,
                "Select action for this message:",
                title,
                JOptionPane.DEFAULT_OPTION,
                JOptionPane.QUESTION_MESSAGE,
                null,
                sendOpts,
                sendOpts[0]
            );
            m.sendMessage(sendOpts[sendChoice]);

            JOptionPane.showMessageDialog(null, m.printMessage(), title,
JOptionPane.INFORMATION_MESSAGE);
            MessageStore.addMessage(m);
        }
    }
}

// Message class with methods for unit-testing
class Message {
    private String id;
    private int number;
    private String recipient;
    private String content;
    private String hash;
    private String status;
    private static int totalCount = 0;

    public Message(String id, int number, String recipient, String content) {
        this.id        = id;
        this.number    = number;
        this.recipient = recipient;
        this.content   = content;
        this.hash      = createMessageHash();
    }

    public boolean checkMessageID() {
        return id != null && id.length() <= 10;
    }

    public boolean checkRecipientCell() {
        return recipient != null && recipient.length() <= 11 && recipient.startsWith("+");
    }

    public String createMessageHash() {
        String idPart = id.substring(0, 2);
        int idx       = number - 1;
        String[] w    = content.trim().split("\\s+");
        String first  = w.length > 0 ? w[0].replaceAll("\\W", "") : "";
        String last   = w.length > 0 ? w[w.length - 1].replaceAll("\\W", "") : "";
        return String.format("%s:%d:%s", idPart, idx, (first + last).toUpperCase());
    }

    public static String generateId() {
        StringBuilder sb = new StringBuilder(10);
        for (int i = 0; i < 10; i++) sb.append((int)(Math.random() * 10));
        return sb.toString();
```

```
    }

    public void sendMessage(String choice) {
        this.status = choice;
        totalCount++;
    }

    public String printMessage() {
        return String.format(
            "ID:%s | Hash:%s | To:%s | \"%s\" | Status:%s",
            id, hash, recipient, content, status
        );
    }

    public int returnTotalMessages() {
        return totalCount;
    }

    // getters for JSON export
    public String getId() { return id; }
    public String getHash() { return hash; }
    public String getRecipient() { return recipient; }
    public String getContent() { return content; }
    public String getStatus() { return status; }
}

// Store and export messages
class MessageStore {
    private static Message[] messages;
    private static int count;
    private static final int INITIAL_CAPACITY = 10;

    public static void init() {
        messages = new Message[INITIAL_CAPACITY];
        count = 0;
    }

    public static void addMessage(Message m) {
        if (count == messages.length) {
            // resize array
            Message[] newArr = new Message[messages.length * 2];
            System.arraycopy(messages, 0, newArr, 0, messages.length);
            messages = newArr;
        }
        messages[count++] = m;
    }
/* Chatgpt JSON Logic */
    public static void exportJson() {
        int export = JOptionPane.showConfirmDialog(
            null,
            "Would you like to export messages to JSON?",
            "Export",
            JOptionPane.YES_NO_OPTION,
            JOptionPane.QUESTION_MESSAGE
        );
        if (export != JOptionPane.YES_OPTION) return;

        JFileChooser chooser = new JFileChooser();
```

```
        chooser.setDialogTitle("Save Messages As");
        chooser.setFileFilter(new FileNameExtensionFilter("JSON Files (*.json)", "json"));
        if (chooser.showSaveDialog(null) != JFileChooser.APPROVE_OPTION) return;

        File file = chooser.getSelectedFile();
        if (!file.getName().toLowerCase().endsWith(".json")) {
            file = new File(file.getAbsolutePath() + ".json");
        }

        try (FileWriter writer = new FileWriter(file)) {
            writer.write("[\n");
            for (int i = 0; i < count; i++) {
                Message m = messages[i];
                writer.write("  {\n");
                writer.write("    \"id\": \"" + m.getId() + "\",\n");
                writer.write("    \"hash\": \"" + m.getHash() + "\",\n");
                writer.write("    \"recipient\": \"" + m.getRecipient() + "\",\n");
                writer.write("    \"content\": \"" + m.getContent().replace("\"", "\\\"") + "\",\n");
                writer.write("    \"status\": \"" + m.getStatus() + "\"\n");
                writer.write("  }" + (i < count - 1 ? "," : "") + "\n");
            }
            writer.write("]");
            JOptionPane.showMessageDialog(
                null,
                "Exported to " + file.getAbsolutePath(),
                "Export Complete",
                JOptionPane.INFORMATION_MESSAGE
            );
        } catch (IOException e) {
            JOptionPane.showMessageDialog(
                null,
                "Export error: " + e.getMessage(),
                "Error",
                JOptionPane.ERROR_MESSAGE
            );
        }
    }
}
```

screenshots

Section B—continued: Junit Tests code-Full Test Code to Test Methods

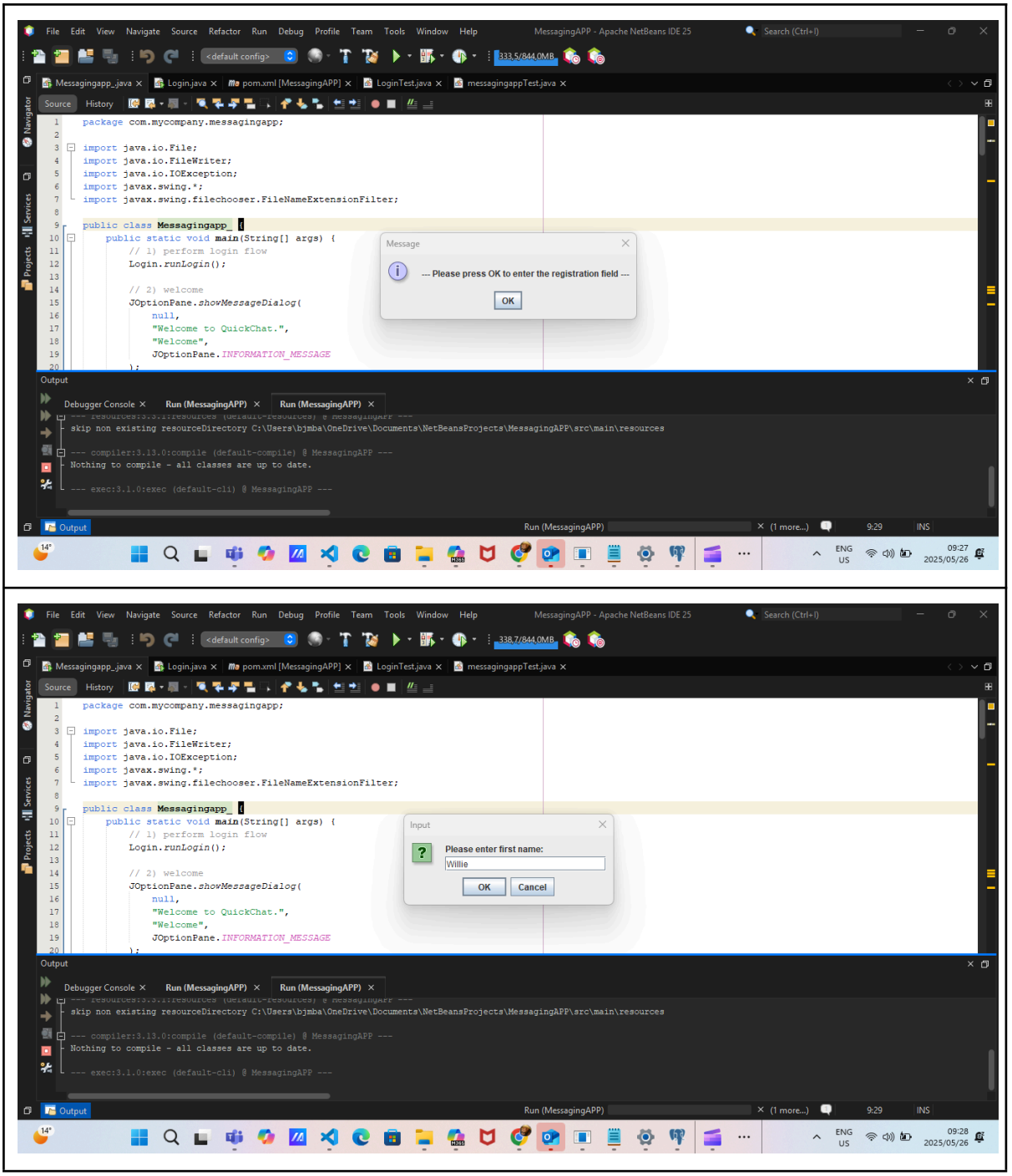
```
package com.mycompany.messagingapp;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class Messagingapp_Test {


    //The tests they specified in the assignment
    @Test
    void testHashForDinnerInvite() {
        Message m1 = new Message(
            "0012345678",
            1,
```

```
        "+277118693002",
        "Hi Mike, can you join us for dinner tonight"
    );

    assertEquals("00:0:HITONIGHT", m1.createMessageHash());}



//Method specific tests,  might be useful for later.
@Test
void testCreateMessageHash() {
    Message m = new Message("AB1234567", 1, "+1234567890", "Hello World");

    assertEquals("AB:0:HELLOWORLD", m.createMessageHash());
}


@Test
void testCheckRecipientCell() {
    Message ok = new Message("ID", 1, "+27831234567", "X");
    assertTrue(ok.checkRecipientCell());
    Message bad = new Message("ID", 1, "27831234567", "X");
    assertFalse(bad.checkRecipientCell());
}

@Test
void testGenerateIdAndCheckMessageID() {
    String id = Message.generateId();
    assertEquals(10, id.length(), "ID must be 10 digits");
    assertTrue(id.matches("\\d{10}"), "ID must be numeric");
    Message m = new Message(id, 1, "+1", "X");
    assertTrue(m.checkMessageID(), "Generated ID should be valid");
}


@Test
void testContentLengthWithinLimit() {
    String shortText = "A".repeat(250);
    assertEquals(250, shortText.length());
    Message m = new Message("ID", 1, "+1", shortText);
    assertTrue(m.getContent().length() <= 250,
            "Content of length 250 should be allowed");
}


@Test
void testContentLengthExceedsLimit() {
    String longText = "A".repeat(251);
    assertEquals(251, longText.length());
    Message m = new Message("ID", 1, "+1", longText);
    assertTrue(m.getContent().length() > 250,
            "Content exceeding 250 chars should be flagged");
}


@Test
void testSendActionSend() {
    Message m = new Message("ID", 1, "+1", "Hi");
```

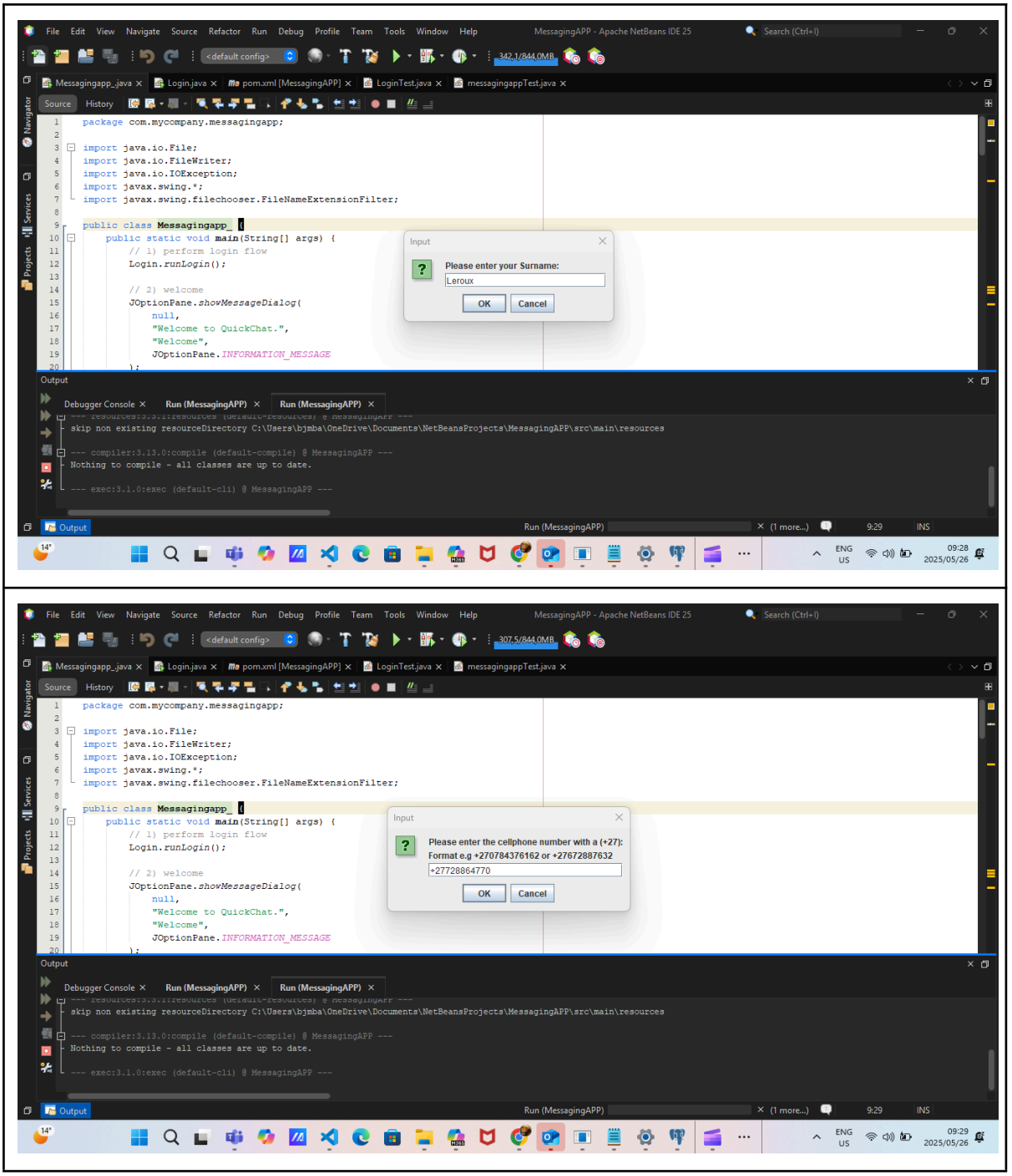
```
        m.sendMessage("Send");
        assertEquals("Send", m.getStatus());
        assertEquals(1, m.returnTotalMessages());
    }


    @Test
    void testSendActionDiscard() {
        Message m = new Message("ID", 1, "+1", "Hi");
        m.sendMessage("Discard");
        assertEquals("Discard", m.getStatus());
        assertEquals(1, m.returnTotalMessages());
    }

    @Test
    void testSendActionStore() {
        Message m = new Message("ID", 1, "+1", "Hi");
        m.sendMessage("Store");
        assertEquals("Store", m.getStatus());
        assertEquals(1, m.returnTotalMessages());
    }

    @Test
    void testTotalCountAfterMultipleSends() {
        Message m1 = new Message("ID1", 1, "+1", "A");
        Message m2 = new Message("ID2", 2, "+1", "B");
        m1.sendMessage("Send");
        m2.sendMessage("Send");
        assertEquals(2, m2.returnTotalMessages(),
                "After sending twice, totalCount should be 2");
    }
}
```

SECTION C: Screenshots of Code Running.

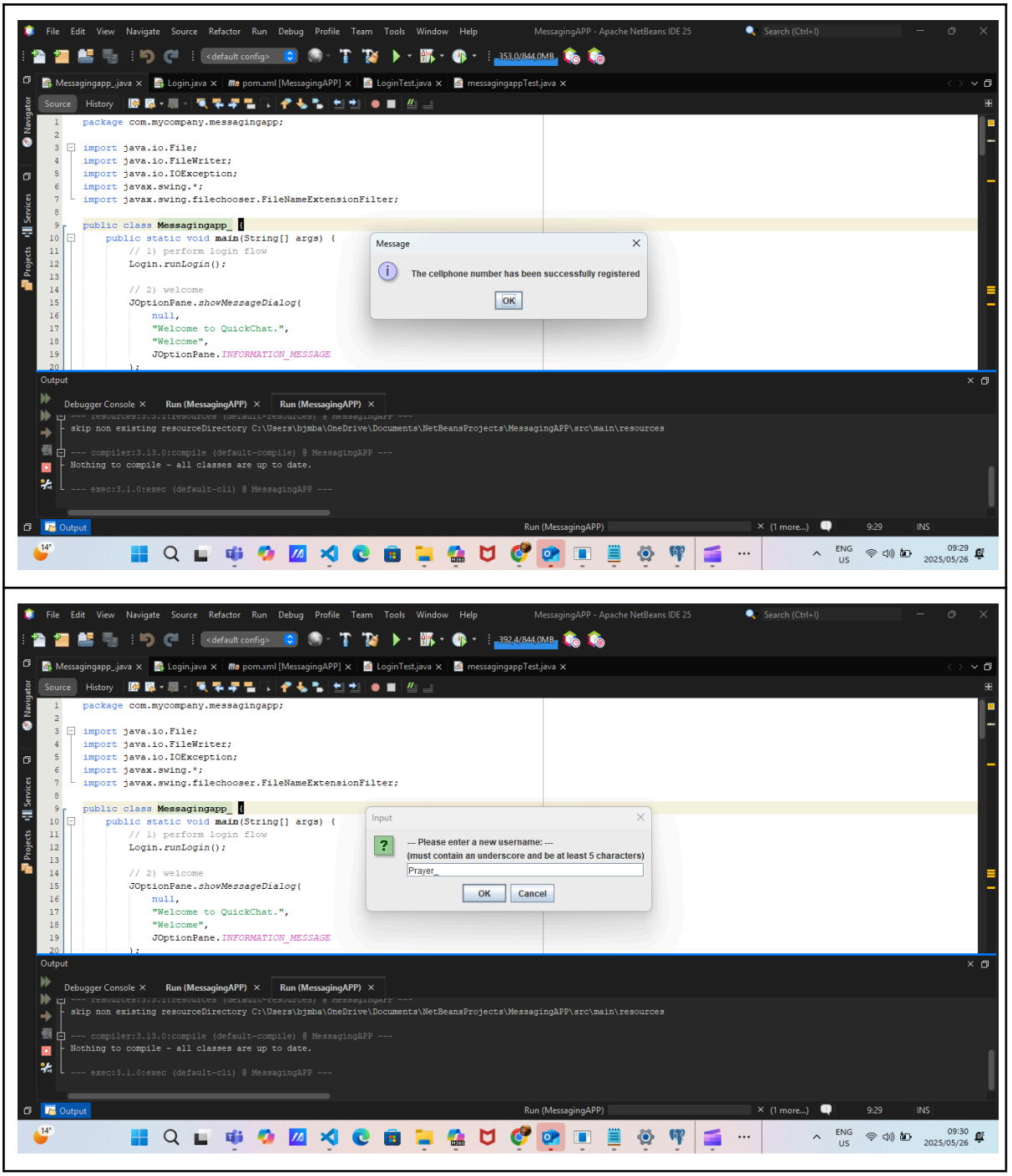
Code running on Netbeans Screenshots:
User Login Logic to the Message class.


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Message
--- Please press OK to enter the registration field ---
OK
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
9:29
INS
09:27
2025/05/26

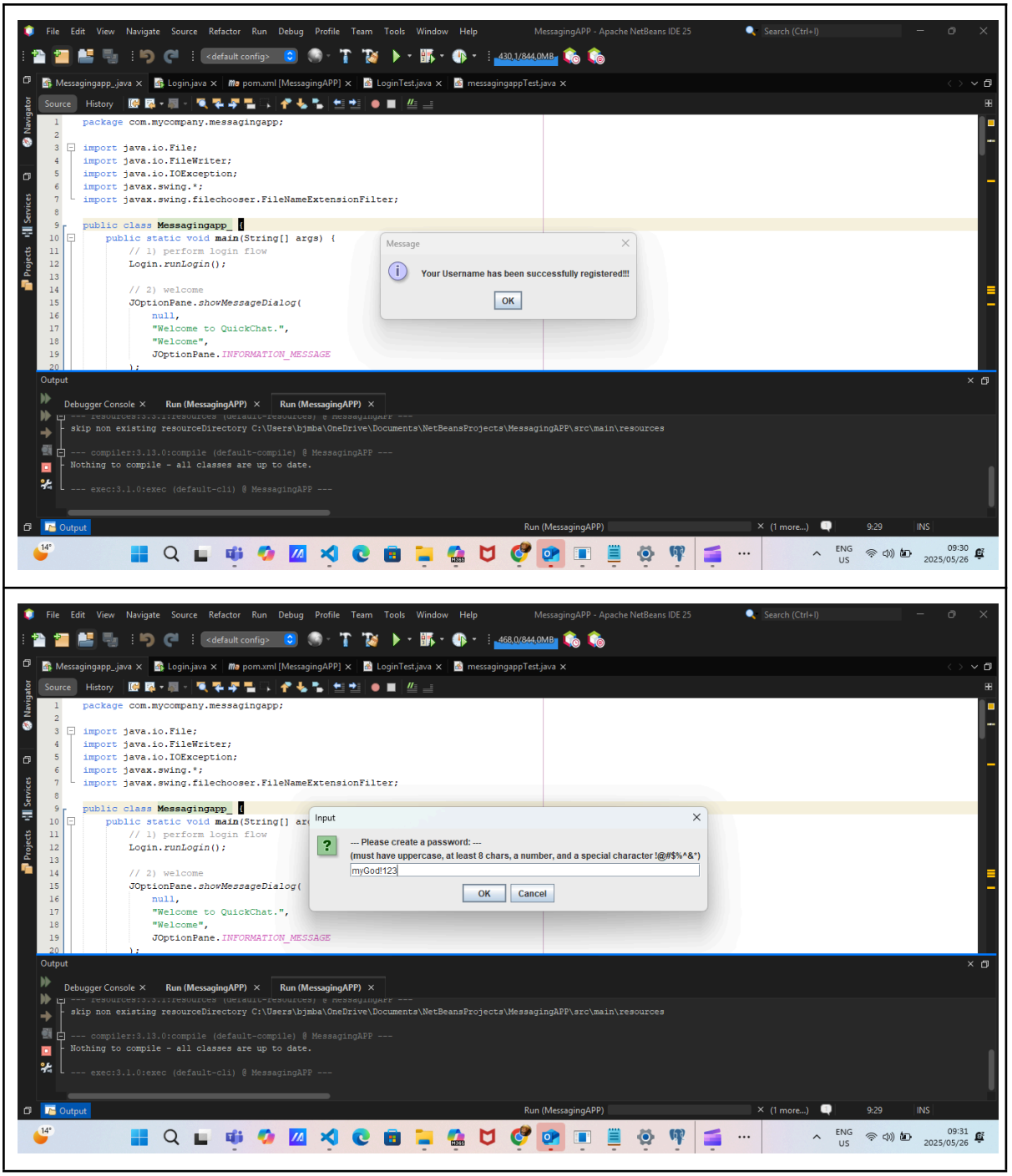
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Input
Please enter first name:
Willie
OK
Cancel
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
9:29
INS
09:28
2025/05/26


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Input
Please enter your Surname:
Leroux
OK
Cancel
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
Output
Run (MessagingAPP)
(1 more...)
9:29
INS
ENG
US
09:28
2025/05/26

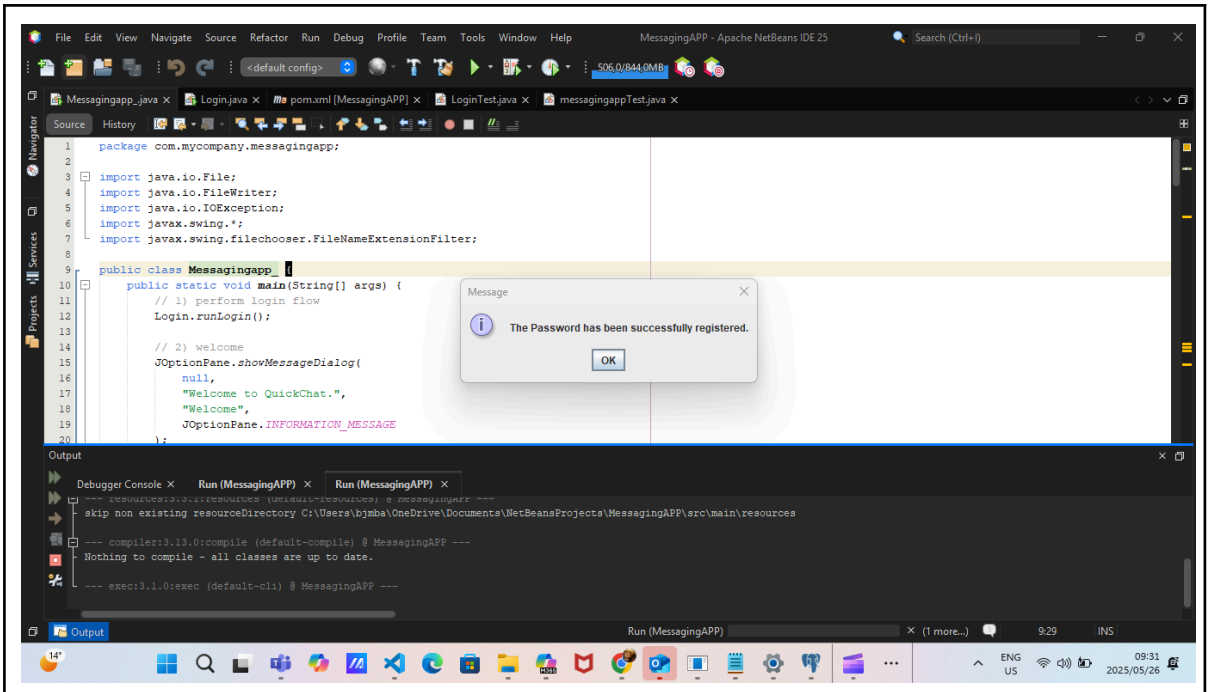
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Input
Please enter the cellphone number with a (+27):
Format e.g +270784376162 or +27672887632
+27728864770
OK
Cancel
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
Output
Run (MessagingAPP)
(1 more...)
9:29
INS
ENG
US
09:29
2025/05/26

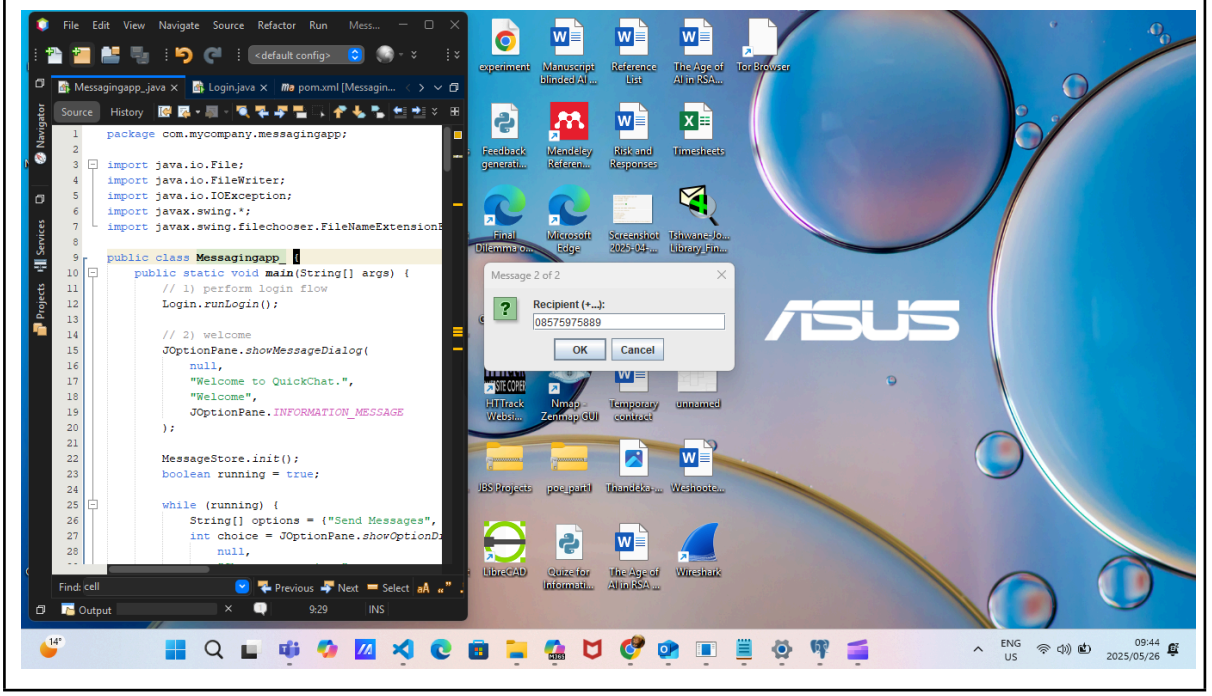

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Message
The cellphone number has been successfully registered
OK
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
Run (MessagingAPP)
(1 more...)
9:29
INS
ENG
US
09:29
2025/05/26

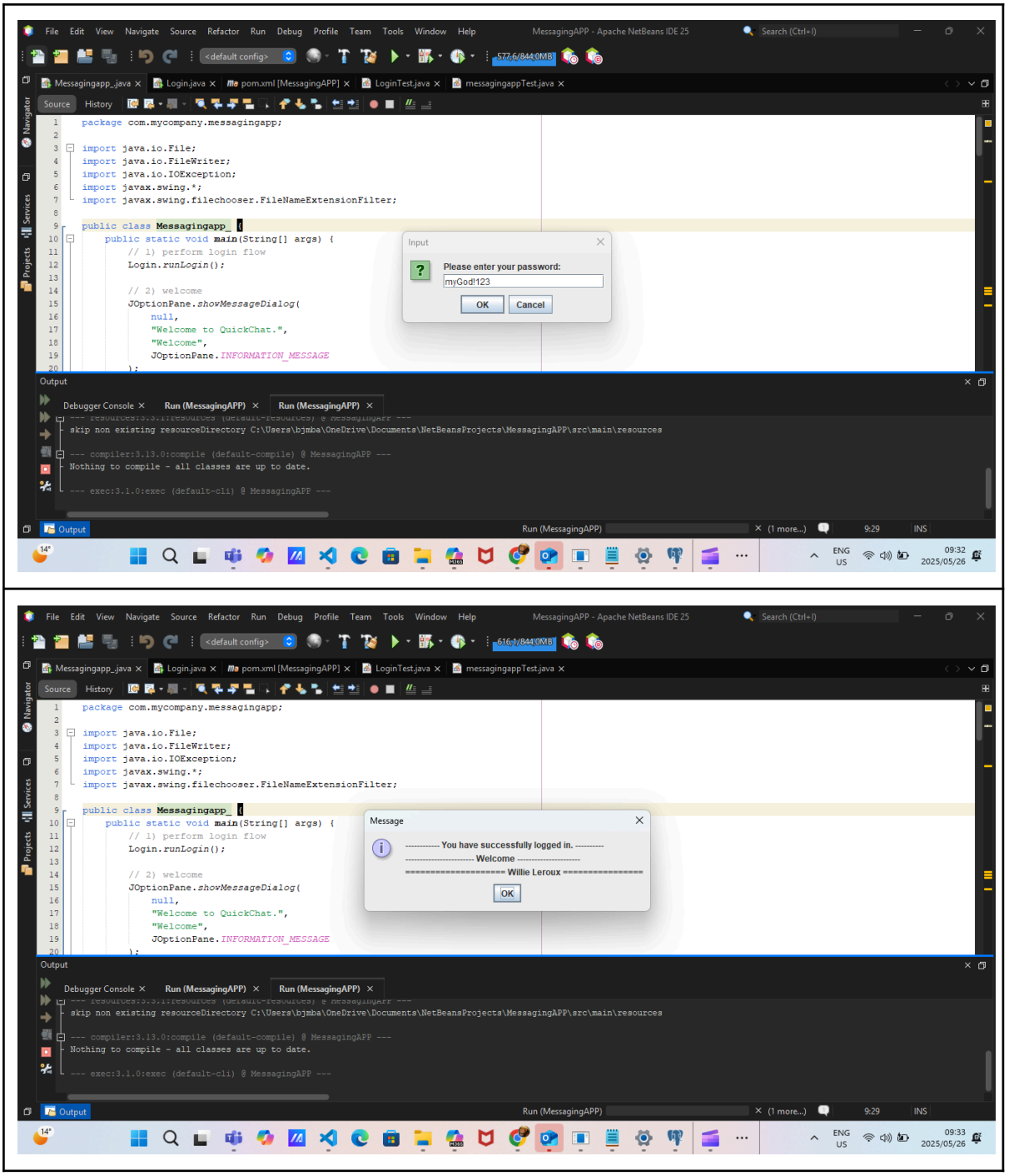
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Input
--- Please enter a new username: ---
(must contain an underscore and be at least 5 characters)
Prayer_
OK
Cancel
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
Run (MessagingAPP)
(1 more...)
9:29
INS
ENG
US
09:30
2025/05/26


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Message
Your Username has been successfully registered!!!
OK
Output
Debugger Console
Run (MessagingAPP)
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
Output
Run (MessagingAPP)
(1 more...)
9:29
INS
ENG
US
09:30
2025/05/26

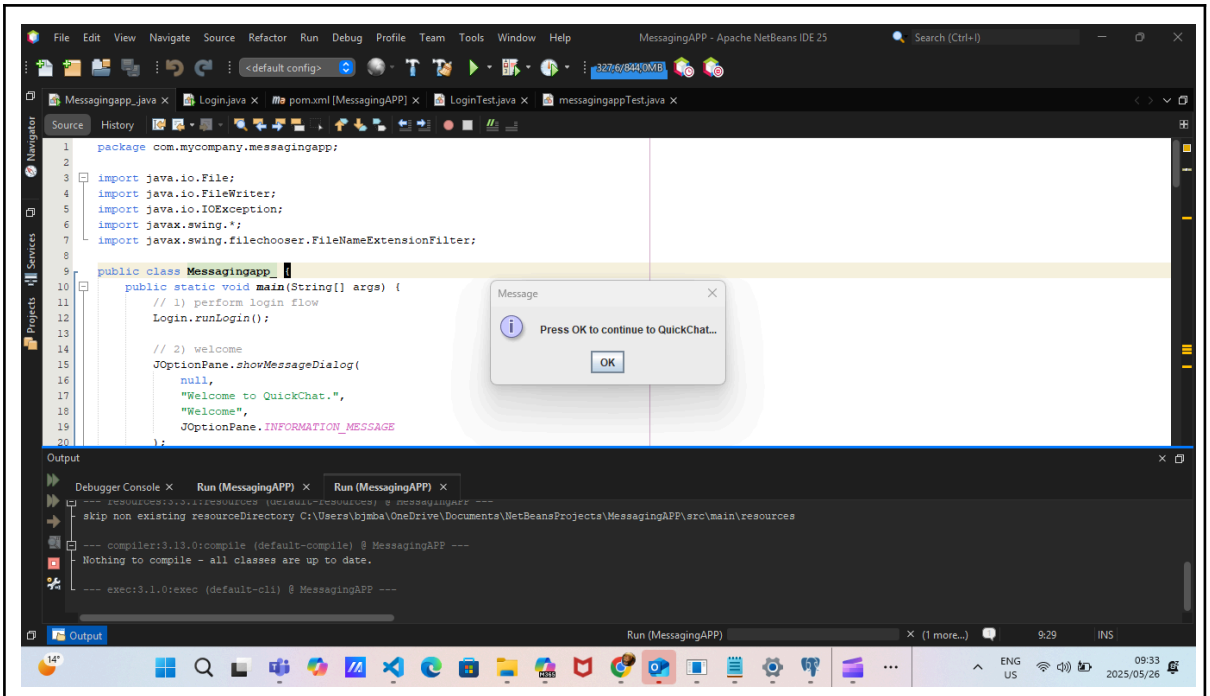
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Input
--- Please create a password: ---
(must have uppercase, at least 8 chars, a number, and a special character !@#$%^&*)
myGod!123
OK
Cancel
Output
Debugger Console
Run (MessagingAPP)
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
Output
Run (MessagingAPP)
(1 more...)
9:29
INS
ENG
US
09:31
2025/05/26

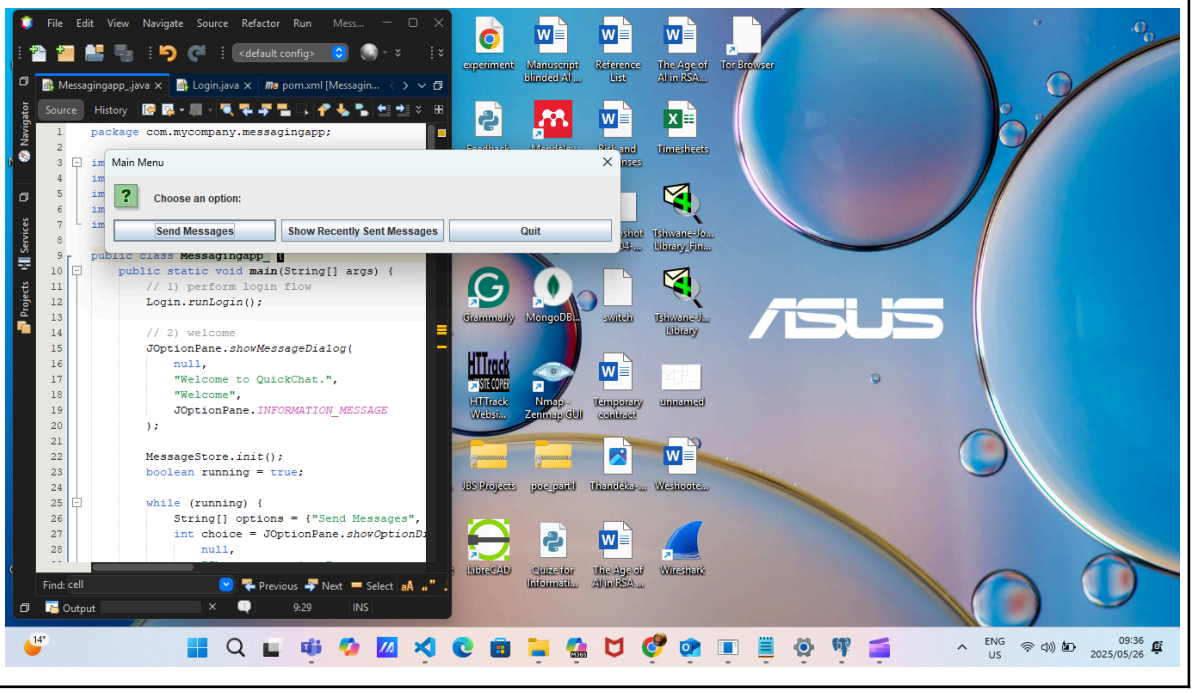

MessagingAPP - Apache NetBeans IDE 25
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Message
The Password has been successfully registered.
OK
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---

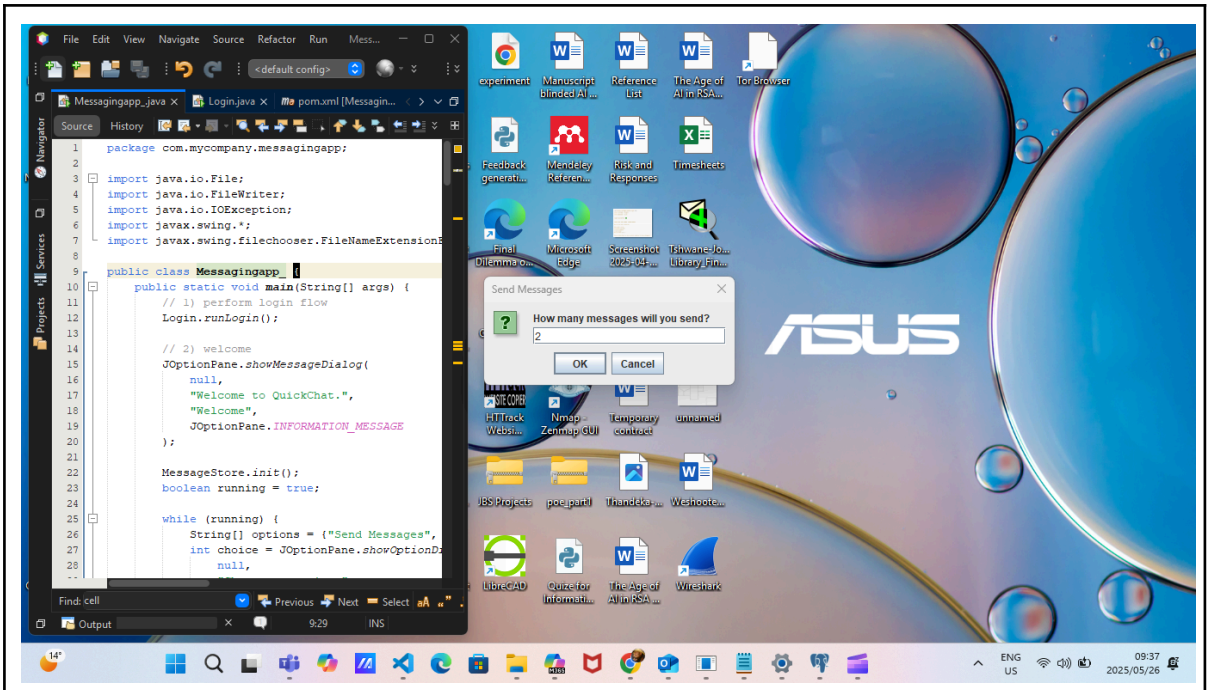

Message 2 of 2
Recipient (+...):
08575975889
OK
Cancel
ASUS

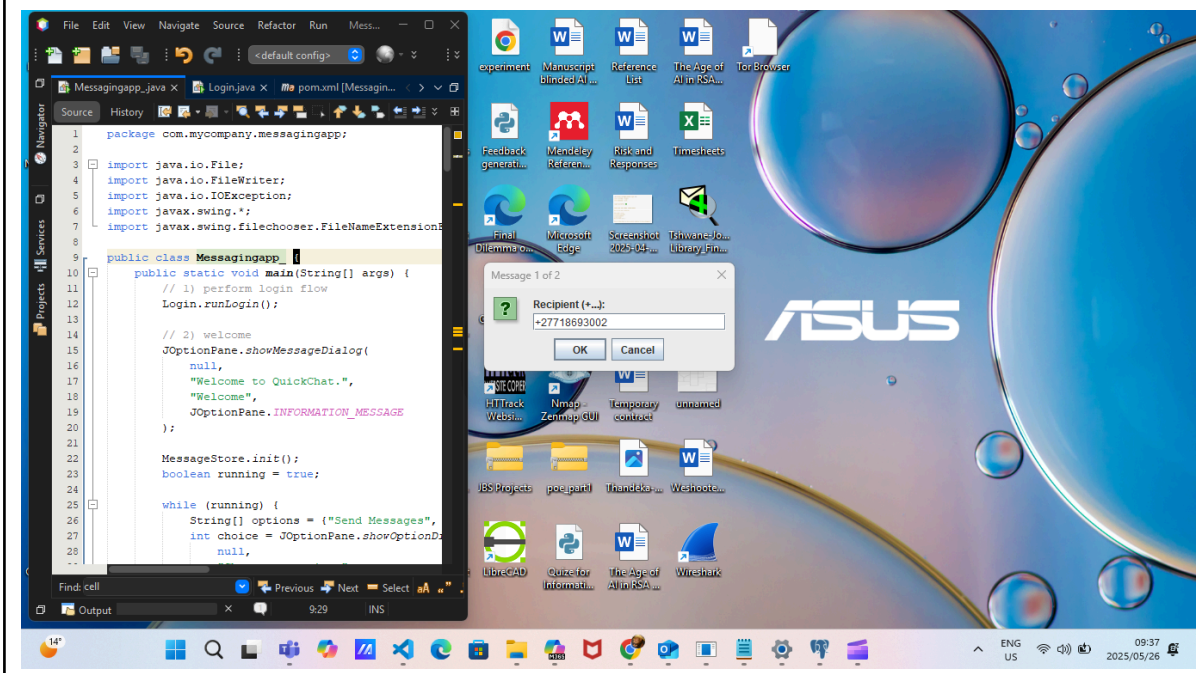

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Input
Please enter your password:
myGod!123
OK
Cancel
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
Run (MessagingAPP)
(1 more...)
9:29
INS
ENG
US
09:32
2025/05/26

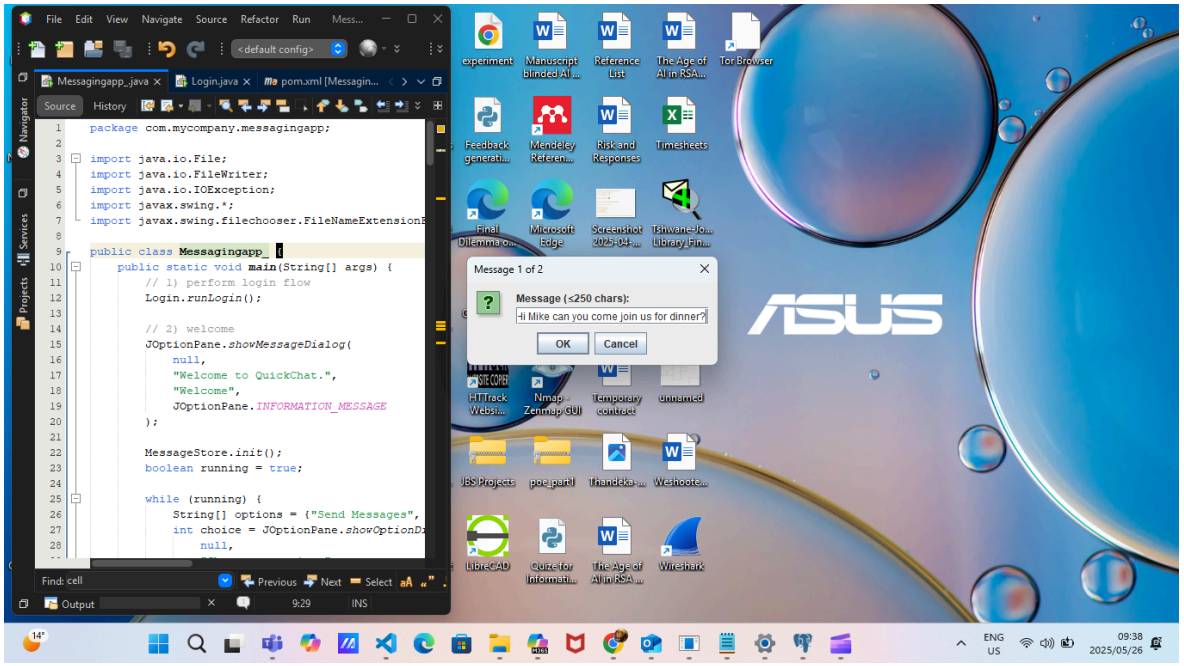
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Message
---------- You have successfully logged in. ----------
---------- Welcome ----------
---------- Willie Leroux ----------
OK
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---
Run (MessagingAPP)
(1 more...)
9:29
INS
ENG
US
09:33
2025/05/26

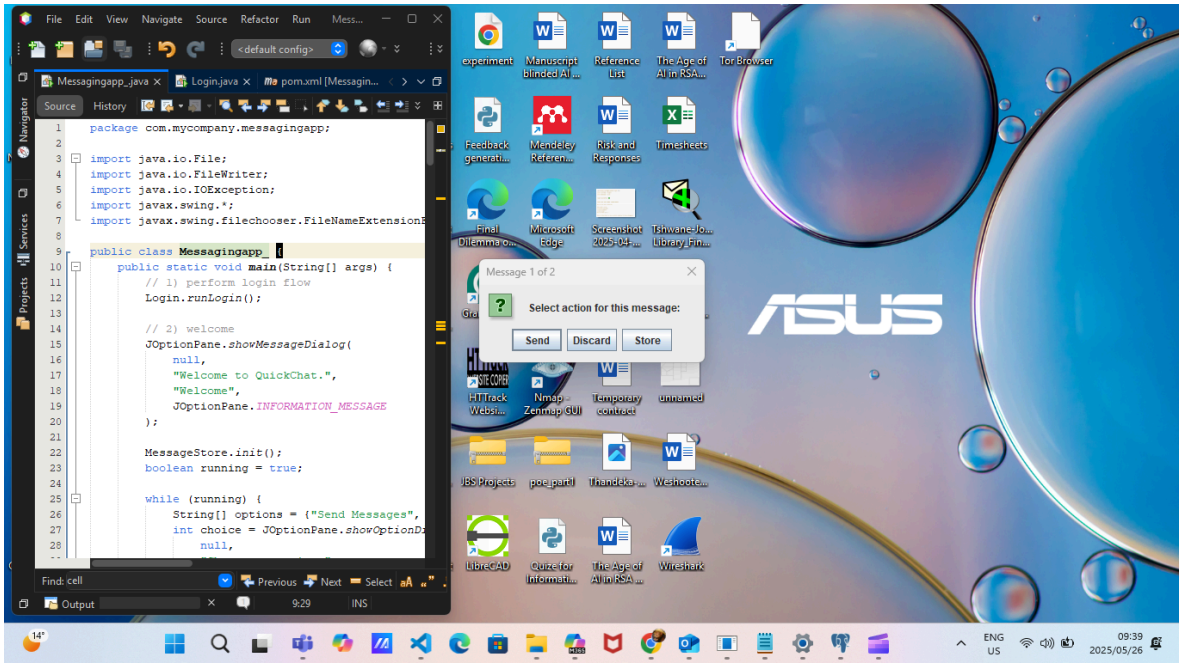

MessagingAPP - Apache NetBeans IDE 25
Messagingapp_java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
Message
Press OK to continue to QuickChat...
OK
Output
Debugger Console
Run (MessagingAPP)
skip non existing resourceDirectory C:\Users\bjmba\OneDrive\Documents\NetBeansProjects\MessagingAPP\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ MessagingAPP ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ MessagingAPP ---


Main Menu
Choose an option:
Send Messages
Show Recently Sent Messages
Quit
experiment
Manuscript Blinded AI ...
Reference List
The Age of AI in RSA...
Tor Browser
Timesheets
Grammarly
MongoDB...
switch
HTTrack Websi...
Nmap - Zenmap GUI
Temporary contract
unnamed
JBS Projects
poe_part1
Thandeka-...
Weshoote...
LibreCAD
Quize for Informati...
The Age of AI in RSA ...
Wireshark
ASUS

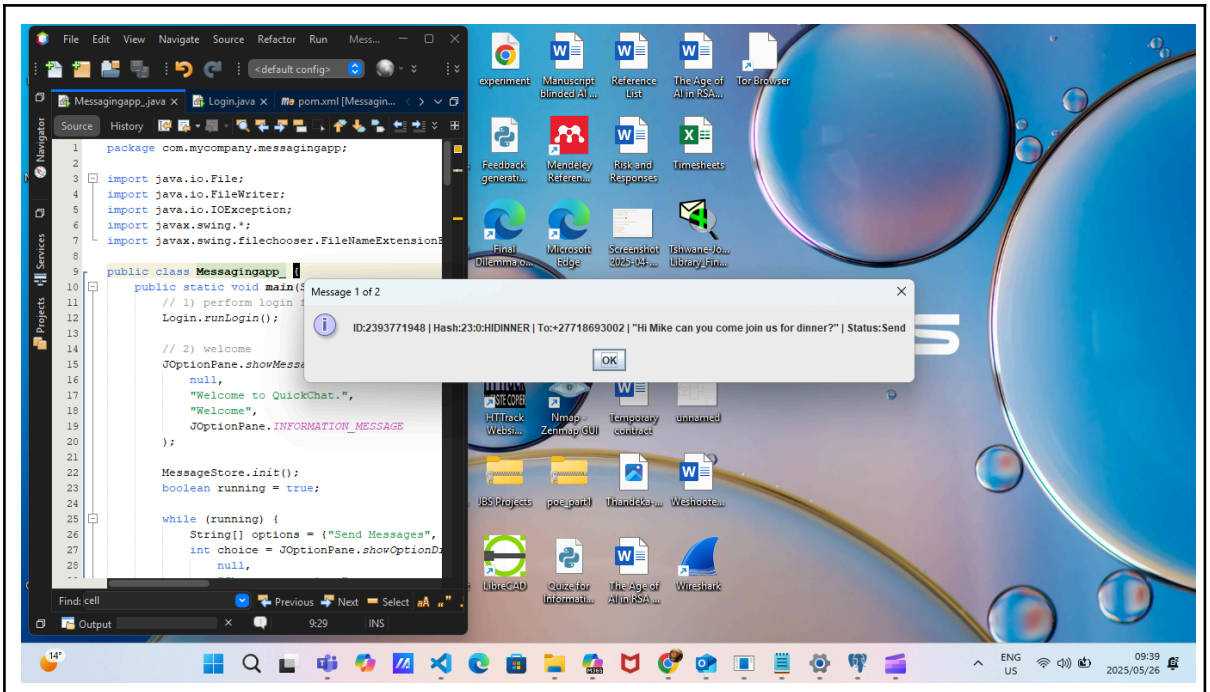

Send Messages
How many messages will you send?
2
OK
Cancel

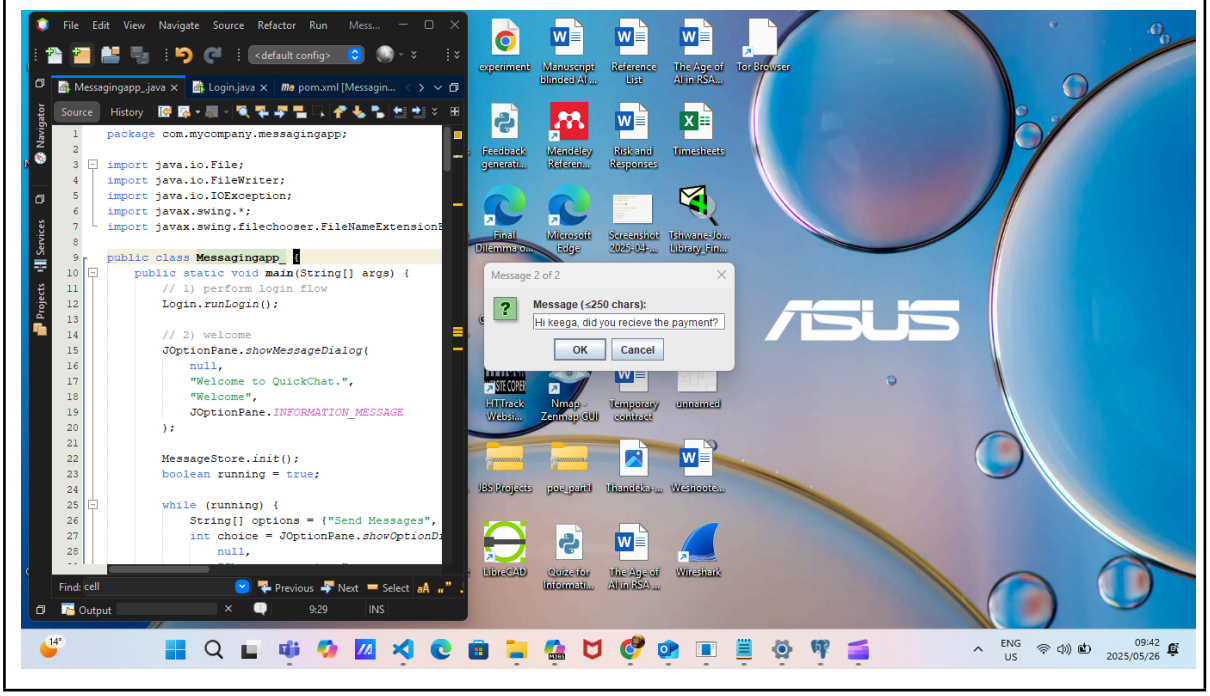

Message 1 of 2
Recipient (+...):
+27718693002
OK
Cancel

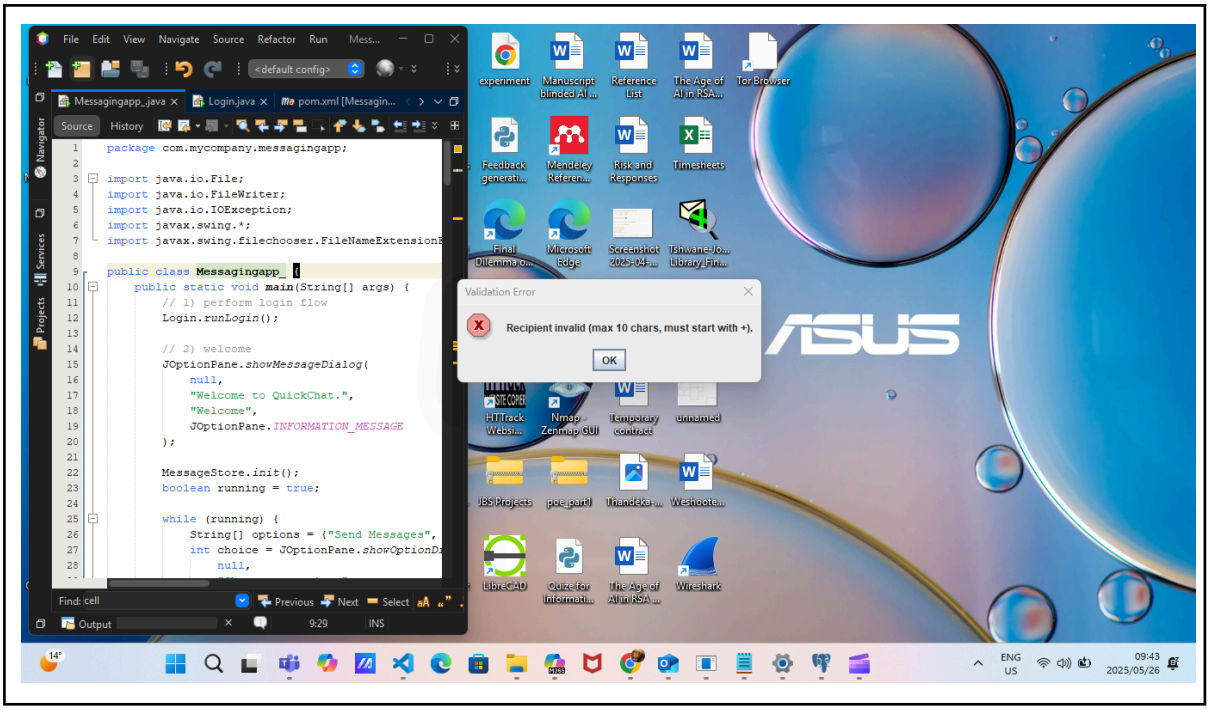

Message 1 of 2
Message (≤250 chars):
Hi Mike can you come join us for dinner?
OK
Cancel

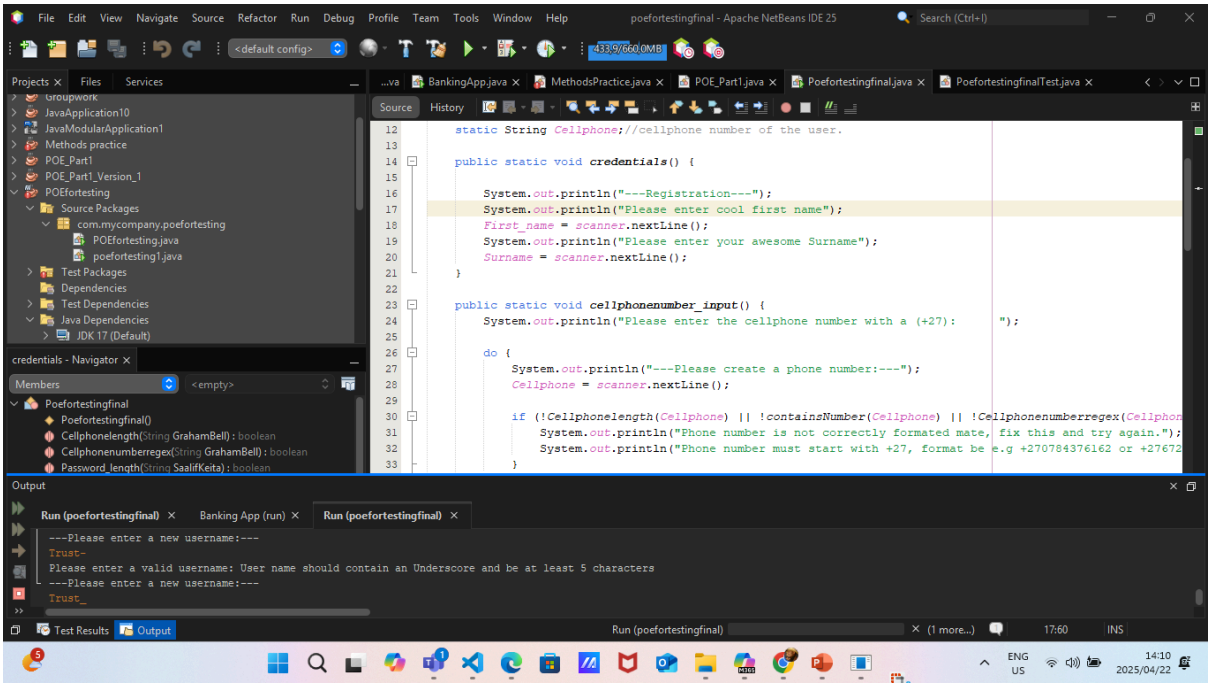

Message 1 of 2
Select action for this message:
Send
Discard
Store

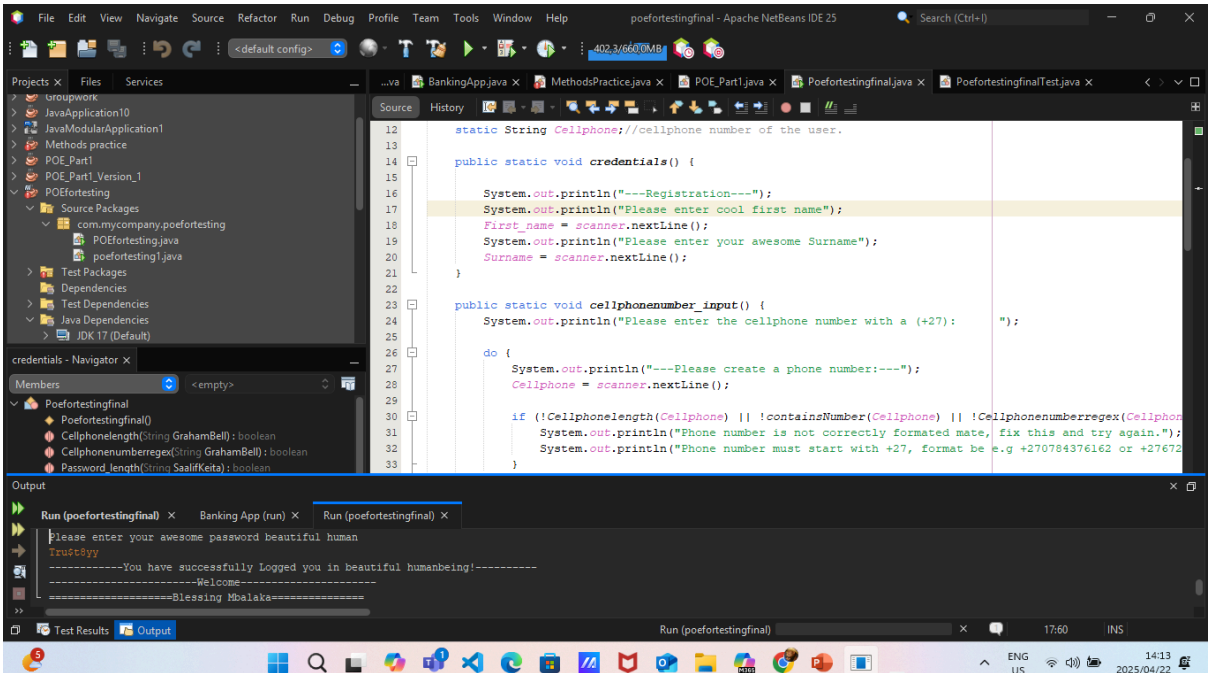
JSON file


Message 1 of 2
ID:2393771948 | Hash:23:0:HIDINNER | To:+27718693002 | "Hi Mike can you come join us for dinner?" | Status:Send
OK


Message 2 of 2
Message (≤250 chars):
Hi keega, did you recieve the payment?
OK
Cancel

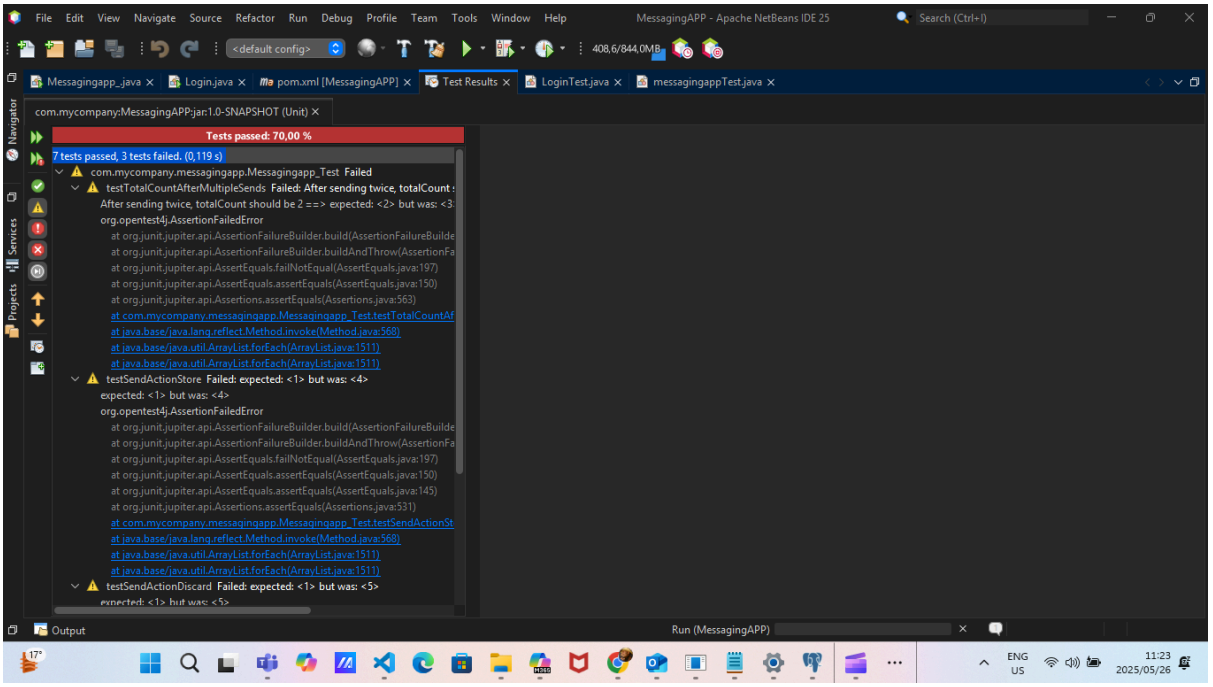

File Edit View Navigate Source Refactor Run
Messagingapp_java
Login.java
pom.xml [Messagin...
Source History
package com.mycompany.messagingapp;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionF
public class Messagingapp_ {
public static void main(String[] args) {
// 1) perform login flow
Login.runLogin();
// 2) welcome
JOptionPane.showMessageDialog(
null,
"Welcome to QuickChat.",
"Welcome",
JOptionPane.INFORMATION_MESSAGE
);
MessageStore.init();
boolean running = true;
while (running) {
String[] options = {"Send Messages",
int choice = JOptionPane.showOptionDi
null,
Find: cell
Previous
Next
Select
Output
9:29
INS
Validation Error
Recipient invalid (max 10 chars, must start with +).
OK
experiment
Manuscript blinded AI ...
Reference List
The Age of AI in RSA...
Tor Browser
Feedback generati...
Mendeley Referen...
Risk and Responses
Timesheets
Final Dilemma o...
Microsoft Edge
Screenshot 2025-04-...
Tshwane-Jo... Library_Fin...
HTTrack Websi...
Nmap - Zenmap GUI
Temporary contract
unnamed
JBS Projects
poe_part1
Thandeka-...
Weshoote...
LibreCAD
Quize for Informati...
The Age of AI in RSA ...
Wireshark
ENG
US
09:43
2025/05/26


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
poefortestingfinal - Apache NetBeans IDE 25
Search (Ctrl+I)
<default config>
Projects
Files
Services
Groupwork
JavaApplication10
JavaModularApplication1
Methods practice
POE_Part1
POE_Part1_Version_1
POEfortesting
Source Packages
com.mycompany.poefortesting
POEfortesting.java
poefortesting1.java
Test Packages
Dependencies
Test Dependencies
Java Dependencies
JDK 17 (Default)
credentials - Navigator
Members
<empty>
Poefortestingfinal
Poefortestingfinal()
Cellphonelength(String GrahamBell) : boolean
Cellphonenumberregex(String GrahamBell) : boolean
Password_length(String SaalifKeita) : boolean
BankingApp.java
MethodsPractice.java
POE_Part1.java
Poefortestingfinal.java
PoefortestingfinalTest.java
Source
History
static String Cellphone;//cellphone number of the user.
public static void credentials() {
System.out.println("---Registration---");
System.out.println("Please enter cool first name");
First_name = scanner.nextLine();
System.out.println("Please enter your awesome Surname");
Surname = scanner.nextLine();
}
public static void cellphonenumber_input() {
System.out.println("Please enter the cellphone number with a (+27): ");
do {
System.out.println("---Please create a phone number:---");
Cellphone = scanner.nextLine();
if (!Cellphonelength(Cellphone) || !containsNumber(Cellphone) || !Cellphonenumberregex(Cellphon
System.out.println("Phone number is not correctly formated mate, fix this and try again.");
System.out.println("Phone number must start with +27, format be e.g +270784376162 or +27672
Output
Run (poefortestingfinal)
Banking App (run)
---Please enter a new username:---
Trust-
Please enter a valid username: User name should contain an Underscore and be at least 5 characters
---Please enter a new username:---
Trust_
Test Results
Output
Run (poefortestingfinal)
(1 more...)
17:60
INS
ENG
US
14:10
2025/04/22

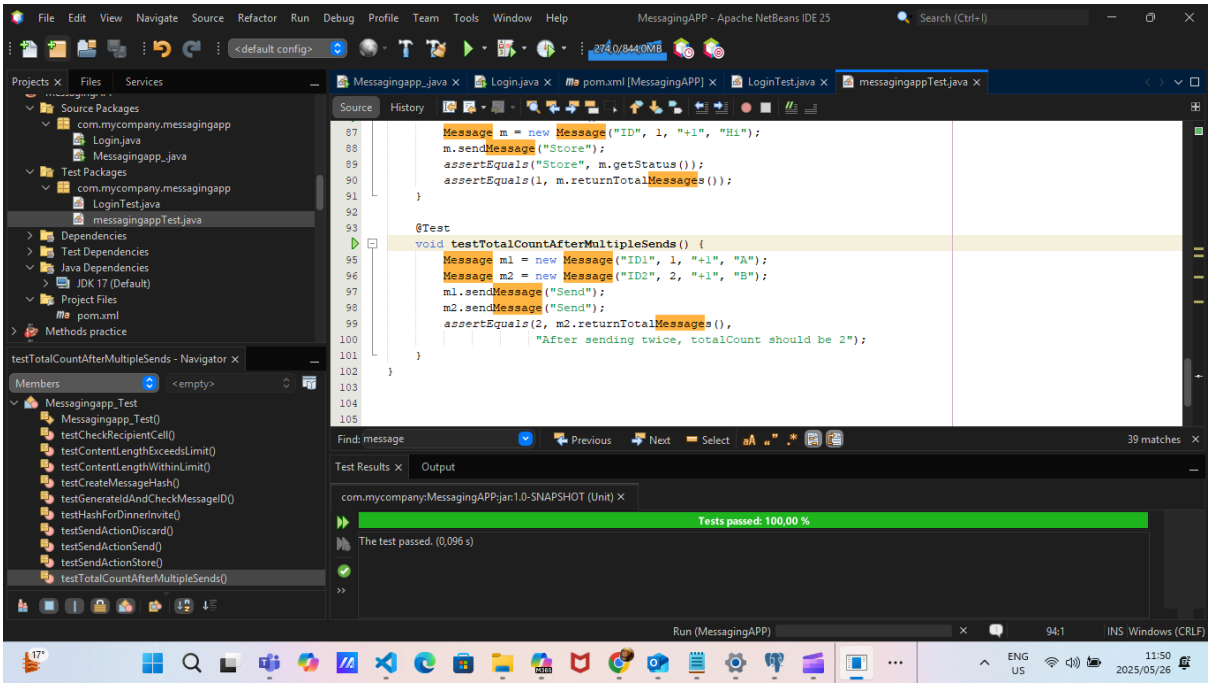

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
poefortestingfinal - Apache NetBeans IDE 25
Search (Ctrl+I)
<default config>
Projects
Files
Services
Groupwork
JavaApplication10
JavaModularApplication1
Methods practice
POE_Part1
POE_Part1_Version_1
POEfortesting
Source Packages
com.mycompany.poefortesting
POEfortesting.java
poefortesting1.java
Test Packages
Dependencies
Test Dependencies
Java Dependencies
JDK 17 (Default)
credentials - Navigator
Members
<empty>
Poefortestingfinal
Poefortestingfinal()
Cellphonelength(String GrahamBell) : boolean
Cellphonenumberregex(String GrahamBell) : boolean
Password_length(String SaalifKeita) : boolean
BankingApp.java
MethodsPractice.java
POE_Part1.java
Poefortestingfinal.java
PoefortestingfinalTest.java
Source
History
static String Cellphone;//cellphone number of the user.
public static void credentials() {
System.out.println("---Registration---");
System.out.println("Please enter cool first name");
First_name = scanner.nextLine();
System.out.println("Please enter your awesome Surname");
Surname = scanner.nextLine();
}
public static void cellphonenumber_input() {
System.out.println("Please enter the cellphone number with a (+27): ");
do {
System.out.println("---Please create a phone number:---");
Cellphone = scanner.nextLine();
if (!Cellphonelength(Cellphone) || !containsNumber(Cellphone) || !Cellphonenumberregex(Cellphon
System.out.println("Phone number is not correctly formated mate, fix this and try again.");
System.out.println("Phone number must start with +27, format be e.g +270784376162 or +27672
Output
Run (poefortestingfinal)
Banking App (run)
Please enter your awesome password beautiful human
Tru$t8yy
----------You have successfully Logged you in beautiful humanbeing!----------
------------------------Welcome------------------------
====================Blessing Mbalaka==============
Test Results
Output
Run (poefortestingfinal)
17:60
INS
ENG
US
14:13
2025/04/22

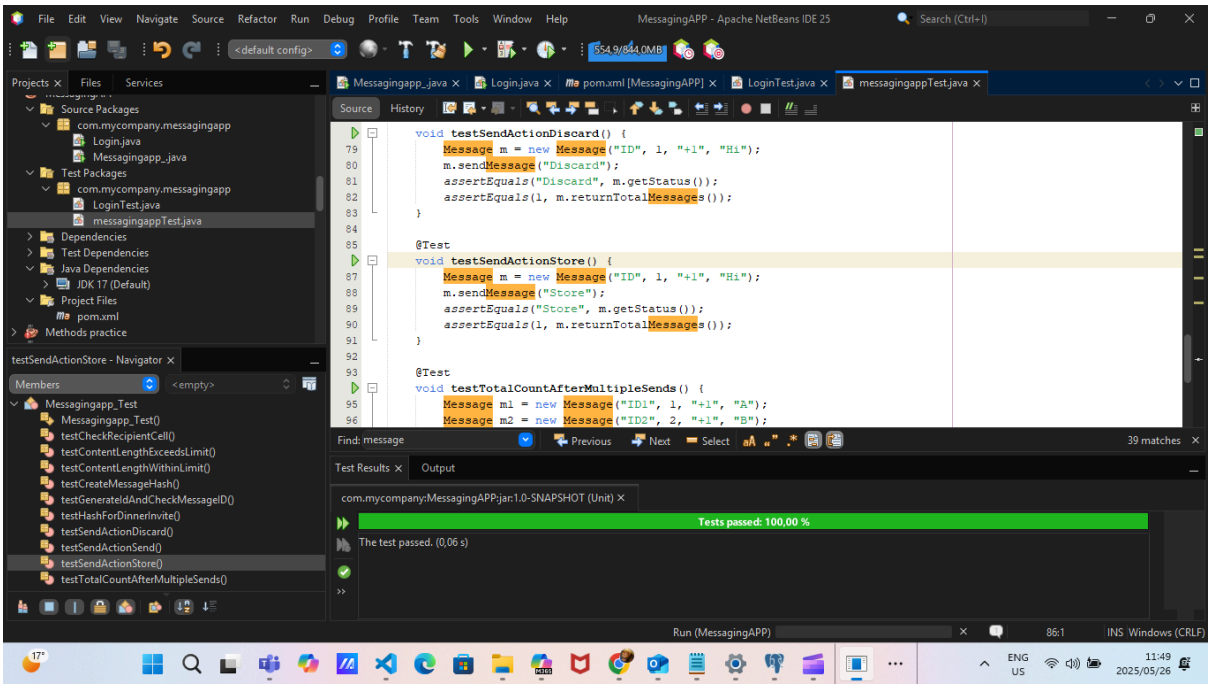
SECTION D – SCREENSHOTS AND OUTCOME OF TESTING of full code

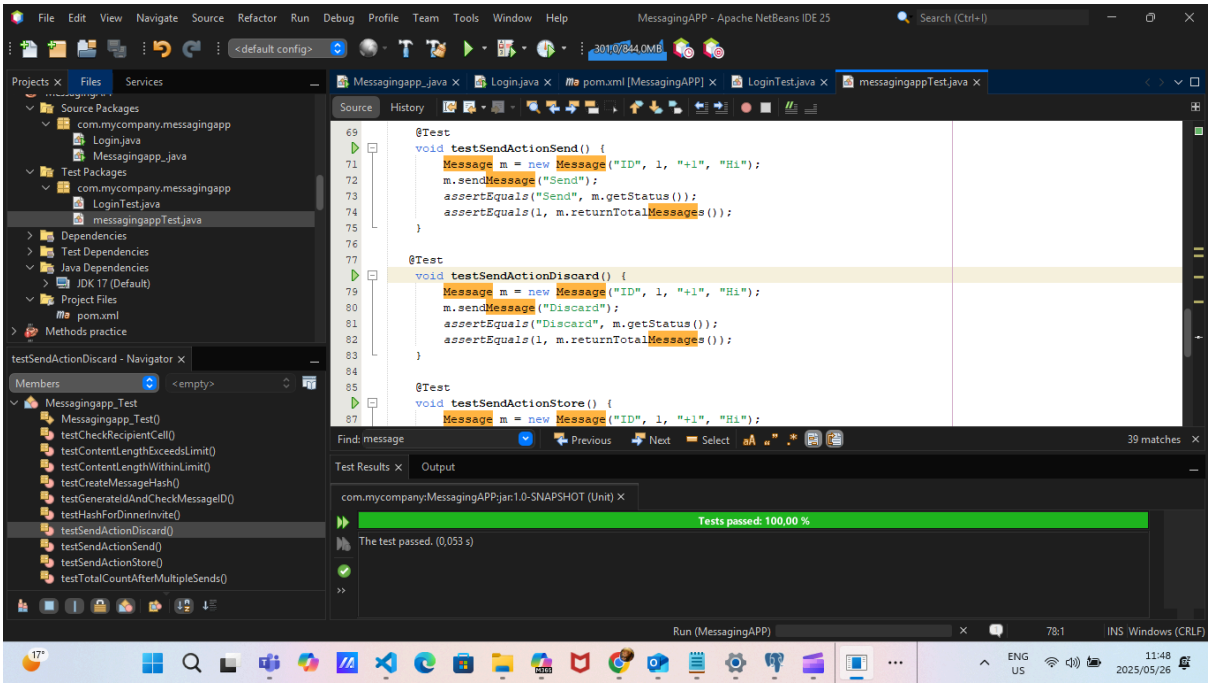
1. **Expanded overall Unit Test**

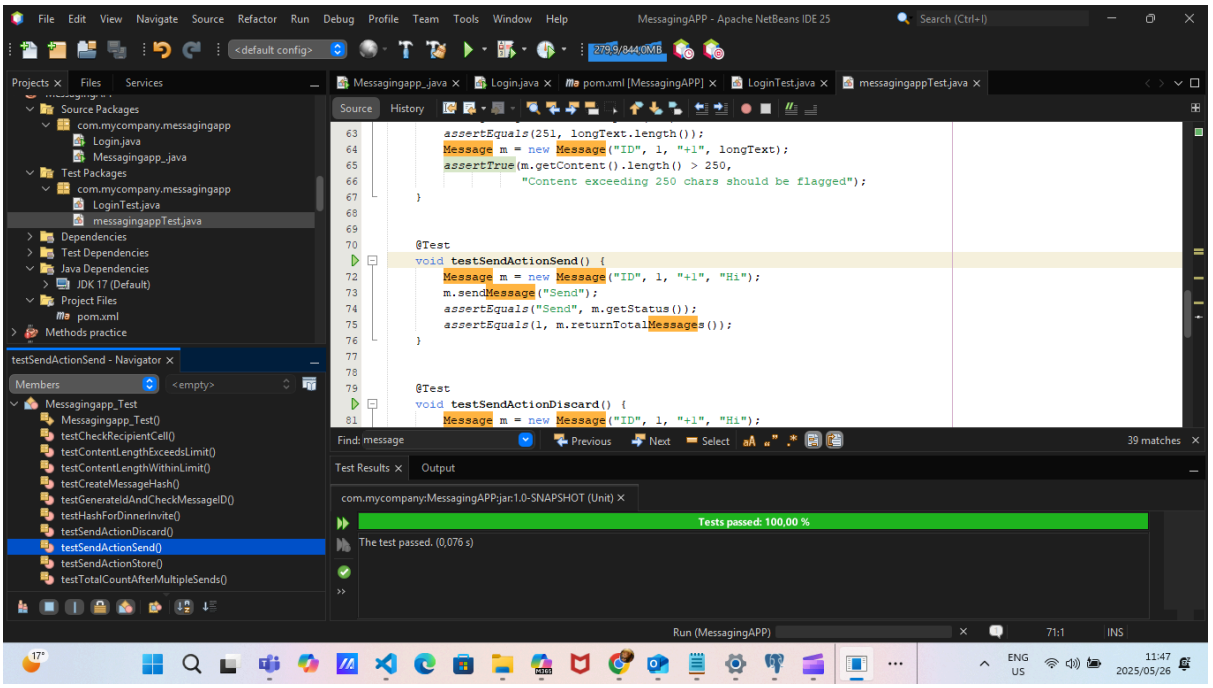


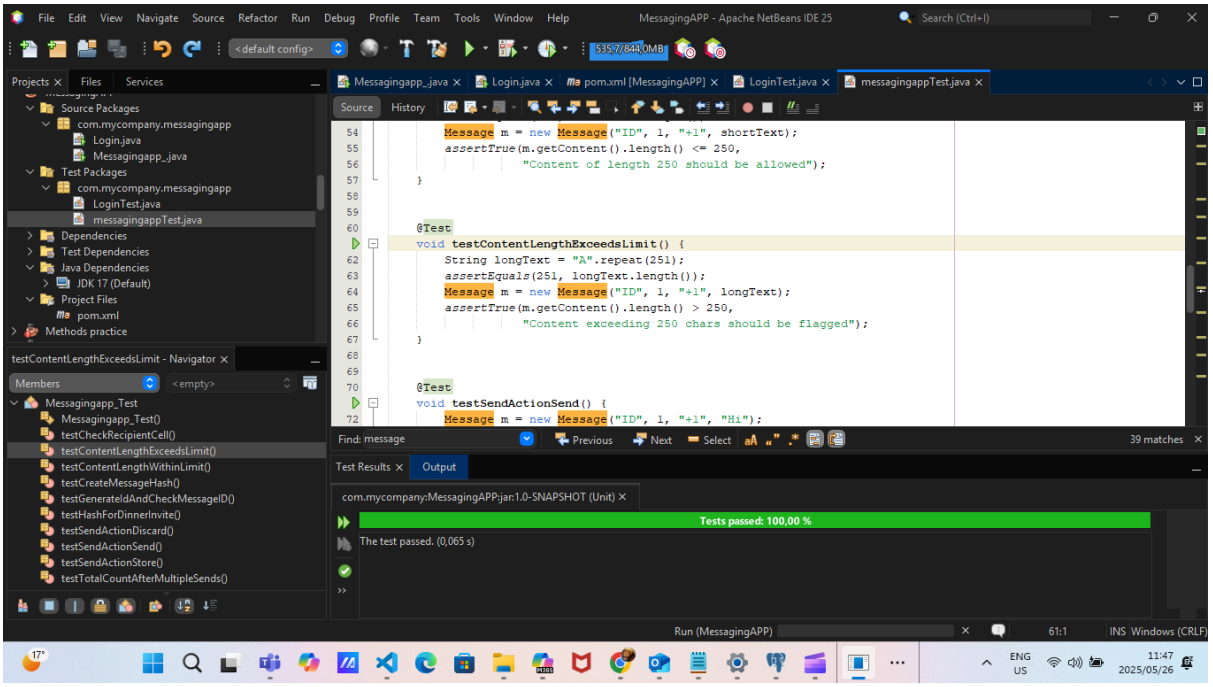
2. **Individual methods of unit tests, non-failed.**

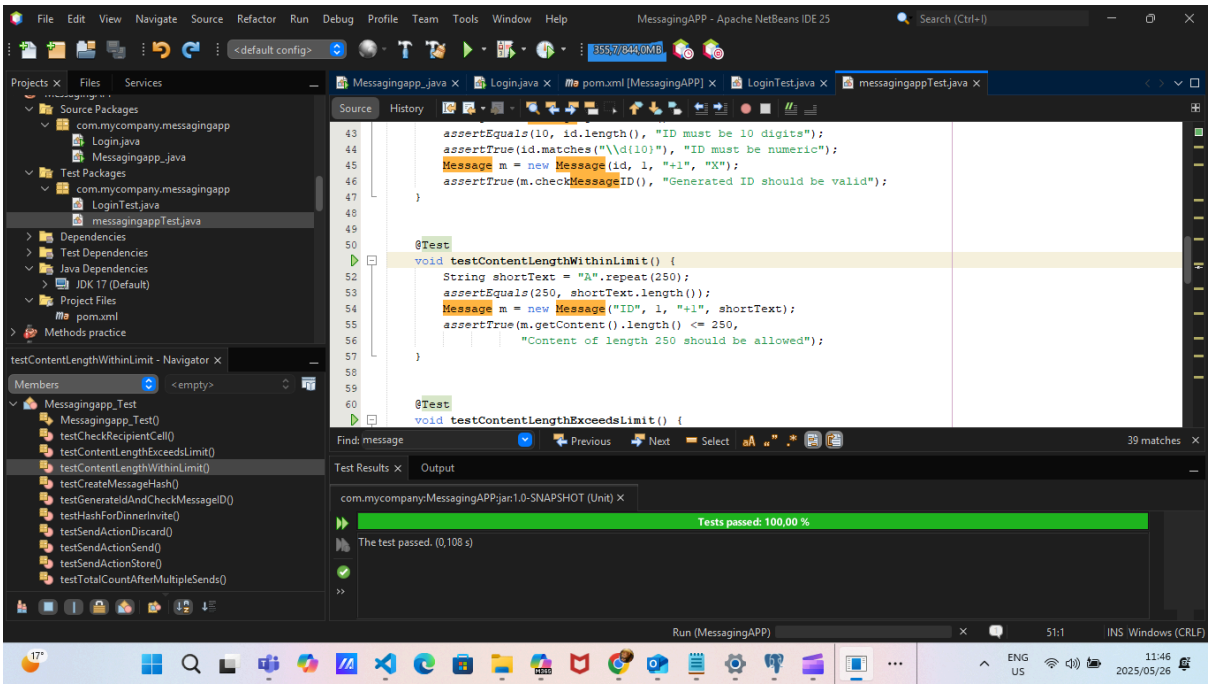

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
<default config>
Projects
Files
Services
Source Packages
com.mycompany.messagingapp
Login.java
Messagingapp_.java
Test Packages
com.mycompany.messagingapp
LoginTest.java
messagingappTest.java
Dependencies
Test Dependencies
Java Dependencies
JDK 17 (Default)
Project Files
pom.xml
Methods practice
testTotalCountAfterMultipleSends - Navigator
Members
<empty>
Messagingapp_Test
Messagingapp_Test()
testCheckRecipientCell()
testContentLengthExceedsLimit()
testContentLengthWithinLimit()
testCreateMessageHash()
testGenerateIdAndCheckMessageID()
testHashForDinnerInvite()
testSendActionDiscard()
testSendActionSend()
testSendActionStore()
testTotalCountAfterMultipleSends()
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
Message m = new Message("ID", 1, "+1", "Hi");
m.sendMessage("Store");
assertEquals("Store", m.getStatus());
assertEquals(1, m.returnTotalMessages());
@Test
void testTotalCountAfterMultipleSends() {
Message m1 = new Message("ID1", 1, "+1", "A");
Message m2 = new Message("ID2", 2, "+1", "B");
m1.sendMessage("Send");
m2.sendMessage("Send");
assertEquals(2, m2.returnTotalMessages(),
"After sending twice, totalCount should be 2");
Find: message
Previous
Next
Select
39 matches
Test Results
Output
com.mycompany:MessagingAPP:jar:1.0-SNAPSHOT (Unit)
Tests passed: 100,00 %
The test passed. (0,096 s)
Run (MessagingAPP)
94:1
INS Windows (CRLF)
ENG
US
11:50
2025/05/26

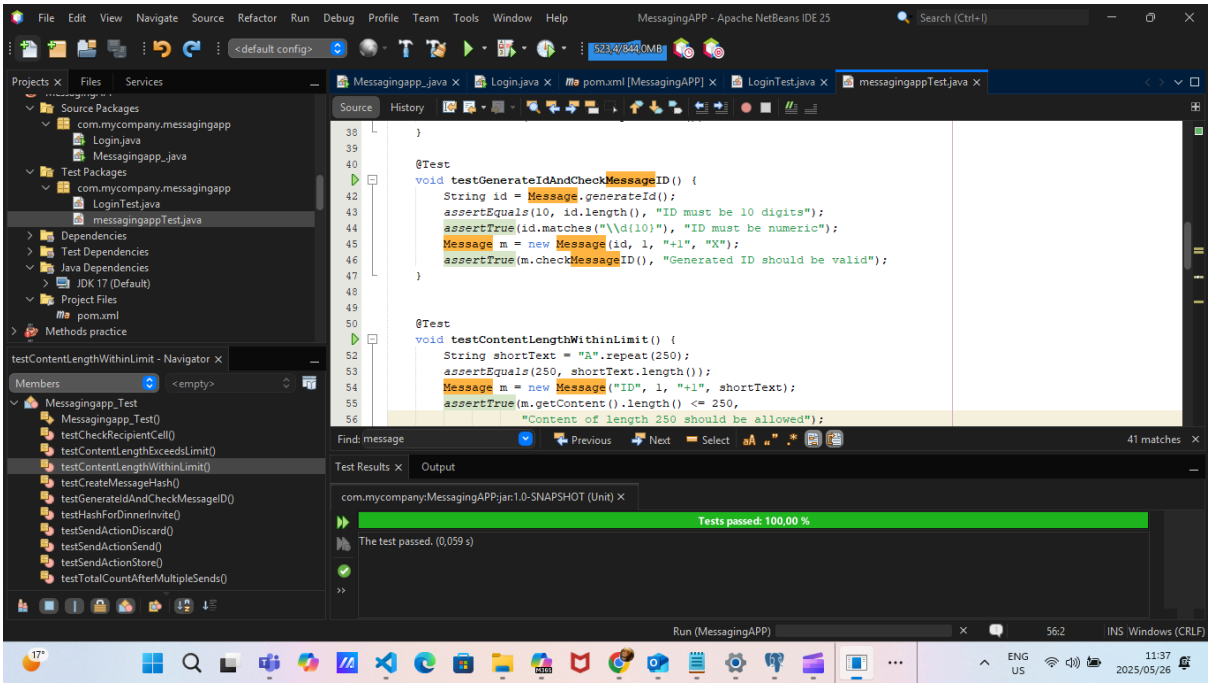

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
<default config>
Projects
Files
Services
Source Packages
com.mycompany.messagingapp
Login.java
Messagingapp_.java
Test Packages
com.mycompany.messagingapp
LoginTest.java
messagingappTest.java
Dependencies
Test Dependencies
Java Dependencies
JDK 17 (Default)
Project Files
pom.xml
Methods practice
testSendActionStore - Navigator
Members
<empty>
Messagingapp_Test
Messagingapp_Test()
testCheckRecipientCell()
testContentLengthExceedsLimit()
testContentLengthWithinLimit()
testCreateMessageHash()
testGenerateIdAndCheckMessageID()
testHashForDinnerInvite()
testSendActionDiscard()
testSendActionSend()
testSendActionStore()
testTotalCountAfterMultipleSends()
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
void testSendActionDiscard() {
Message m = new Message("ID", 1, "+1", "Hi");
m.sendMessage("Discard");
assertEquals("Discard", m.getStatus());
assertEquals(1, m.returnTotalMessages());
@Test
void testSendActionStore() {
Message m = new Message("ID", 1, "+1", "Hi");
m.sendMessage("Store");
assertEquals("Store", m.getStatus());
assertEquals(1, m.returnTotalMessages());
@Test
void testTotalCountAfterMultipleSends() {
Message m1 = new Message("ID1", 1, "+1", "A");
Message m2 = new Message("ID2", 2, "+1", "B");
Find: message
Previous
Next
Select
39 matches
Test Results
Output
com.mycompany:MessagingAPP:jar:1.0-SNAPSHOT (Unit)
Tests passed: 100,00 %
The test passed. (0,06 s)
Run (MessagingAPP)
86:1
INS Windows (CRLF)
ENG
US
11:49
2025/05/26

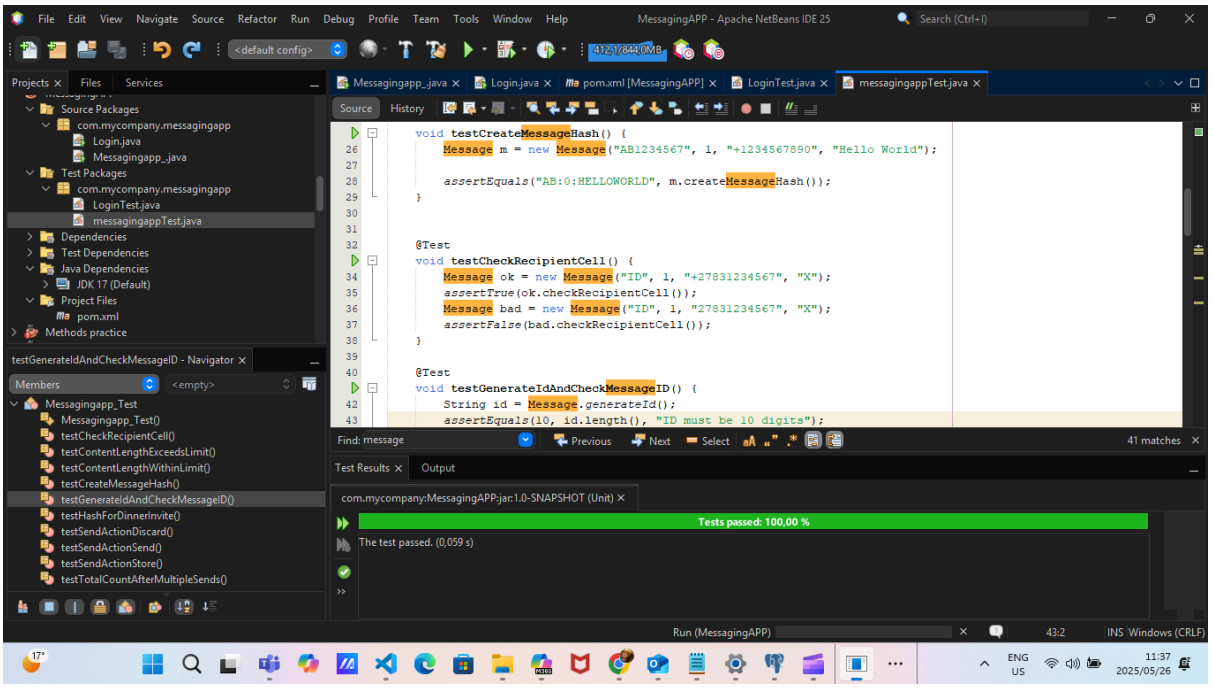

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
<default config>
Projects
Files
Services
Source Packages
com.mycompany.messagingapp
Login.java
Messagingapp_.java
Test Packages
com.mycompany.messagingapp
LoginTest.java
messagingappTest.java
Dependencies
Test Dependencies
Java Dependencies
JDK 17 (Default)
Project Files
pom.xml
Methods practice
testSendActionDiscard - Navigator
Members
<empty>
Messagingapp_Test
Messagingapp_Test()
testCheckRecipientCell()
testContentLengthExceedsLimit()
testContentLengthWithinLimit()
testCreateMessageHash()
testGenerateIdAndCheckMessageID()
testHashForDinnerInvite()
testSendActionDiscard()
testSendActionSend()
testSendActionStore()
testTotalCountAfterMultipleSends()
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
@Test
void testSendActionSend() {
Message m = new Message("ID", 1, "+1", "Hi");
m.sendMessage("Send");
assertEquals("Send", m.getStatus());
assertEquals(1, m.returnTotalMessages());
}
@Test
void testSendActionDiscard() {
Message m = new Message("ID", 1, "+1", "Hi");
m.sendMessage("Discard");
assertEquals("Discard", m.getStatus());
assertEquals(1, m.returnTotalMessages());
}
@Test
void testSendActionStore() {
Message m = new Message("ID", 1, "+1", "Hi");
Find: message
Previous
Next
Select
39 matches
Test Results
Output
com.mycompany:MessagingAPP:jar:1.0-SNAPSHOT (Unit)
Tests passed: 100,00 %
The test passed. (0,053 s)
Run (MessagingAPP)
78:1
INS
Windows (CRLF)
ENG
US
11:48
2025/05/26

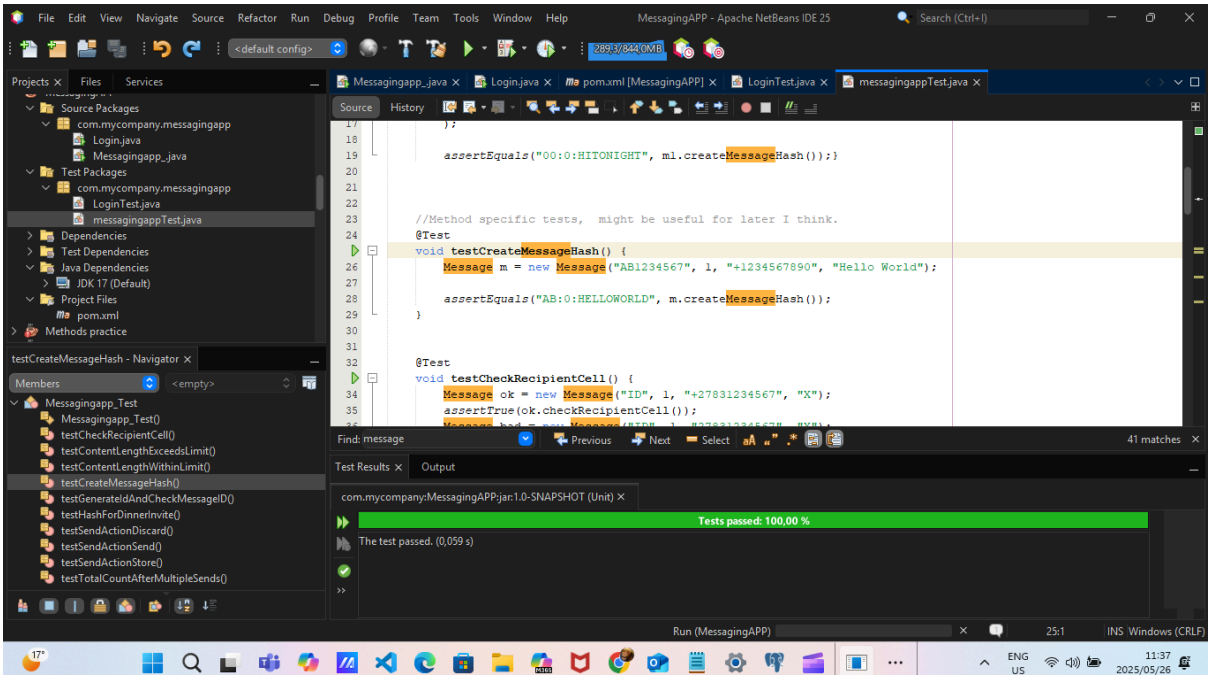

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
<default config>
Projects
Files
Services
Source Packages
com.mycompany.messagingapp
Login.java
Messagingapp_.java
Test Packages
com.mycompany.messagingapp
LoginTest.java
messagingappTest.java
Dependencies
Test Dependencies
Java Dependencies
JDK 17 (Default)
Project Files
pom.xml
Methods practice
testSendActionSend - Navigator
Members
<empty>
Messagingapp_Test
Messagingapp_Test()
testCheckRecipientCell()
testContentLengthExceedsLimit()
testContentLengthWithinLimit()
testCreateMessageHash()
testGenerateIdAndCheckMessageID()
testHashForDinnerInvite()
testSendActionDiscard()
testSendActionSend()
testSendActionStore()
testTotalCountAfterMultipleSends()
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
assertEquals(251, longText.length());
Message m = new Message("ID", 1, "+1", longText);
assertTrue(m.getContent().length() > 250,
"Content exceeding 250 chars should be flagged");
}
@Test
void testSendActionSend() {
Message m = new Message("ID", 1, "+1", "Hi");
m.sendMessage("Send");
assertEquals("Send", m.getStatus());
assertEquals(1, m.returnTotalMessages());
}
@Test
void testSendActionDiscard() {
Message m = new Message("ID", 1, "+1", "Hi");
Find: message
Previous
Next
Select
39 matches
Test Results
Output
com.mycompany:MessagingAPP:jar:1.0-SNAPSHOT (Unit)
Tests passed: 100,00 %
The test passed. (0,076 s)
Run (MessagingAPP)
71:1
INS
ENG
US
11:47
2025/05/26

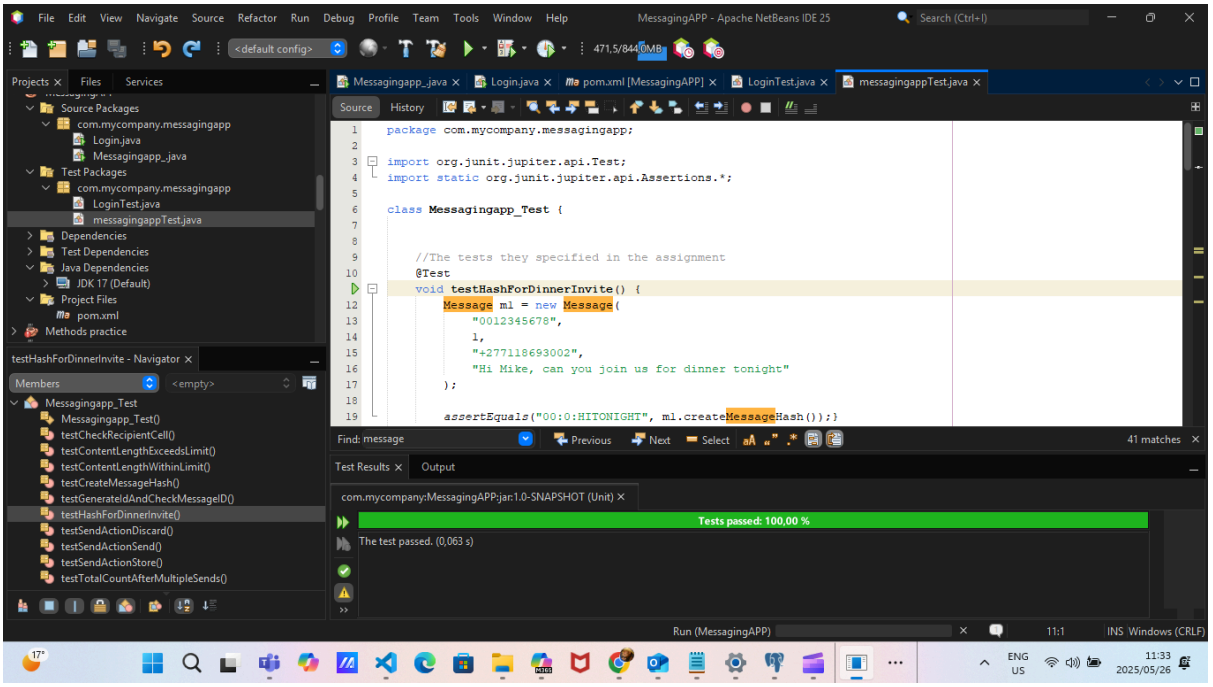

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
<default config>
535,7/844,0MB
Projects
Files
Services
Source Packages
com.mycompany.messagingapp
Login.java
Messagingapp_.java
Test Packages
com.mycompany.messagingapp
LoginTest.java
messagingappTest.java
Dependencies
Test Dependencies
Java Dependencies
JDK 17 (Default)
Project Files
pom.xml
Methods practice
testContentLengthExceedsLimit - Navigator
Members
<empty>
Messagingapp_Test
Messagingapp_Test()
testCheckRecipientCell()
testContentLengthExceedsLimit()
testContentLengthWithinLimit()
testCreateMessageHash()
testGenerateIdAndCheckMessageID()
testHashForDinnerInvite()
testSendActionDiscard()
testSendActionSend()
testSendActionStore()
testTotalCountAfterMultipleSends()
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
Message m = new Message("ID", 1, "+1", shortText);
assertTrue(m.getContent().length() <= 250,
"Content of length 250 should be allowed");
}
@Test
void testContentLengthExceedsLimit() {
String longText = "A".repeat(251);
assertEquals(251, longText.length());
Message m = new Message("ID", 1, "+1", longText);
assertTrue(m.getContent().length() > 250,
"Content exceeding 250 chars should be flagged");
}
@Test
void testSendActionSend() {
Message m = new Message("ID", 1, "+1", "Hi");
Find: message
Previous
Next
Select
39 matches
Test Results
Output
com.mycompany:MessagingAPP:jar:1.0-SNAPSHOT (Unit)
Tests passed: 100,00 %
The test passed. (0,065 s)
Run (MessagingAPP)
61:1
INS Windows (CRLF)
ENG
US
11:47
2025/05/26


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
MessagingAPP - Apache NetBeans IDE 25
Search (Ctrl+I)
<default config>
355,7/844,0MB
Projects
Files
Services
Source Packages
com.mycompany.messagingapp
Login.java
Messagingapp_.java
Test Packages
com.mycompany.messagingapp
LoginTest.java
messagingappTest.java
Dependencies
Test Dependencies
Java Dependencies
JDK 17 (Default)
Project Files
pom.xml
Methods practice
testContentLengthWithinLimit - Navigator
Members
<empty>
Messagingapp_Test
Messagingapp_Test()
testCheckRecipientCell()
testContentLengthExceedsLimit()
testContentLengthWithinLimit()
testCreateMessageHash()
testGenerateIdAndCheckMessageID()
testHashForDinnerInvite()
testSendActionDiscard()
testSendActionSend()
testSendActionStore()
testTotalCountAfterMultipleSends()
Messagingapp_.java
Login.java
pom.xml [MessagingAPP]
LoginTest.java
messagingappTest.java
Source
History
assertEquals(10, id.length(), "ID must be 10 digits");
assertTrue(id.matches("\\d{10}"), "ID must be numeric");
Message m = new Message(id, 1, "+1", "X");
assertTrue(m.checkMessageID(), "Generated ID should be valid");
}
@Test
void testContentLengthWithinLimit() {
String shortText = "A".repeat(250);
assertEquals(250, shortText.length());
Message m = new Message("ID", 1, "+1", shortText);
assertTrue(m.getContent().length() <= 250,
"Content of length 250 should be allowed");
}
@Test
void testContentLengthExceedsLimit() {
Find: message
Previous
Next
Select
39 matches
Test Results
Output
com.mycompany:MessagingAPP:jar:1.0-SNAPSHOT (Unit)
Tests passed: 100,00 %
The test passed. (0,108 s)
Run (MessagingAPP)
51:1
INS Windows (CRLF)
ENG
US
11:46
2025/05/26


MessagingAPP - Apache NetBeans IDE 25
testContentLengthWithinLimit - Navigator
testContentLengthWithinLimit()
Tests passed: 100,00 %
The test passed. (0,059 s)


MessagingAPP - Apache NetBeans IDE 25
testGenerateIdAndCheckMessageID - Navigator
testGenerateIdAndCheckMessageID()
Tests passed: 100,00 %
The test passed. (0,059 s)





Bibliography

Farrell, J. 2023. *Java programming*. 10th ed. Boston: Cengage.

OpenAI. 2025. Chat-GPT (Version 4.0). [Large language model]. Available at: https://chat.openai.com/ [Accessed: 20 March 2024].