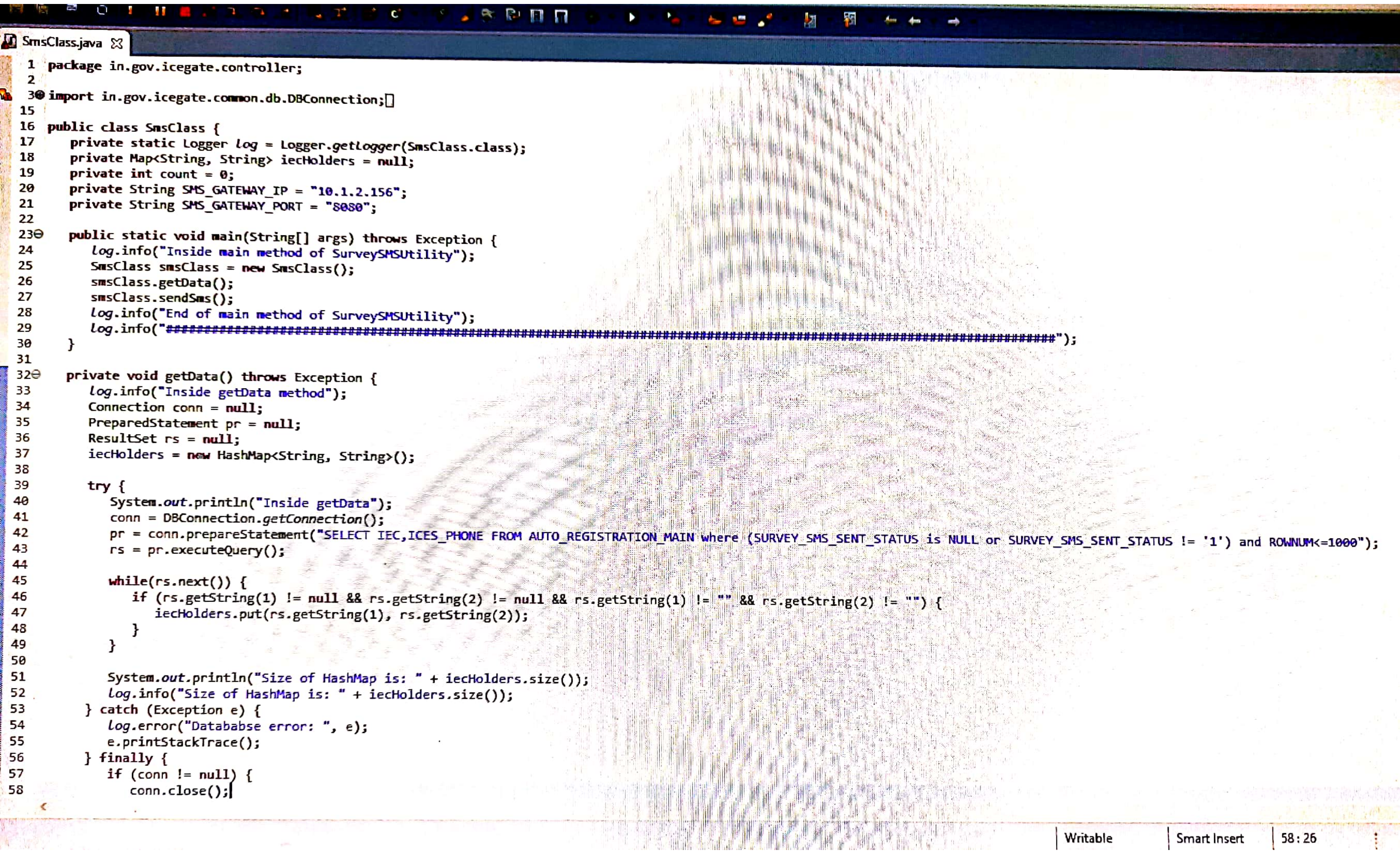
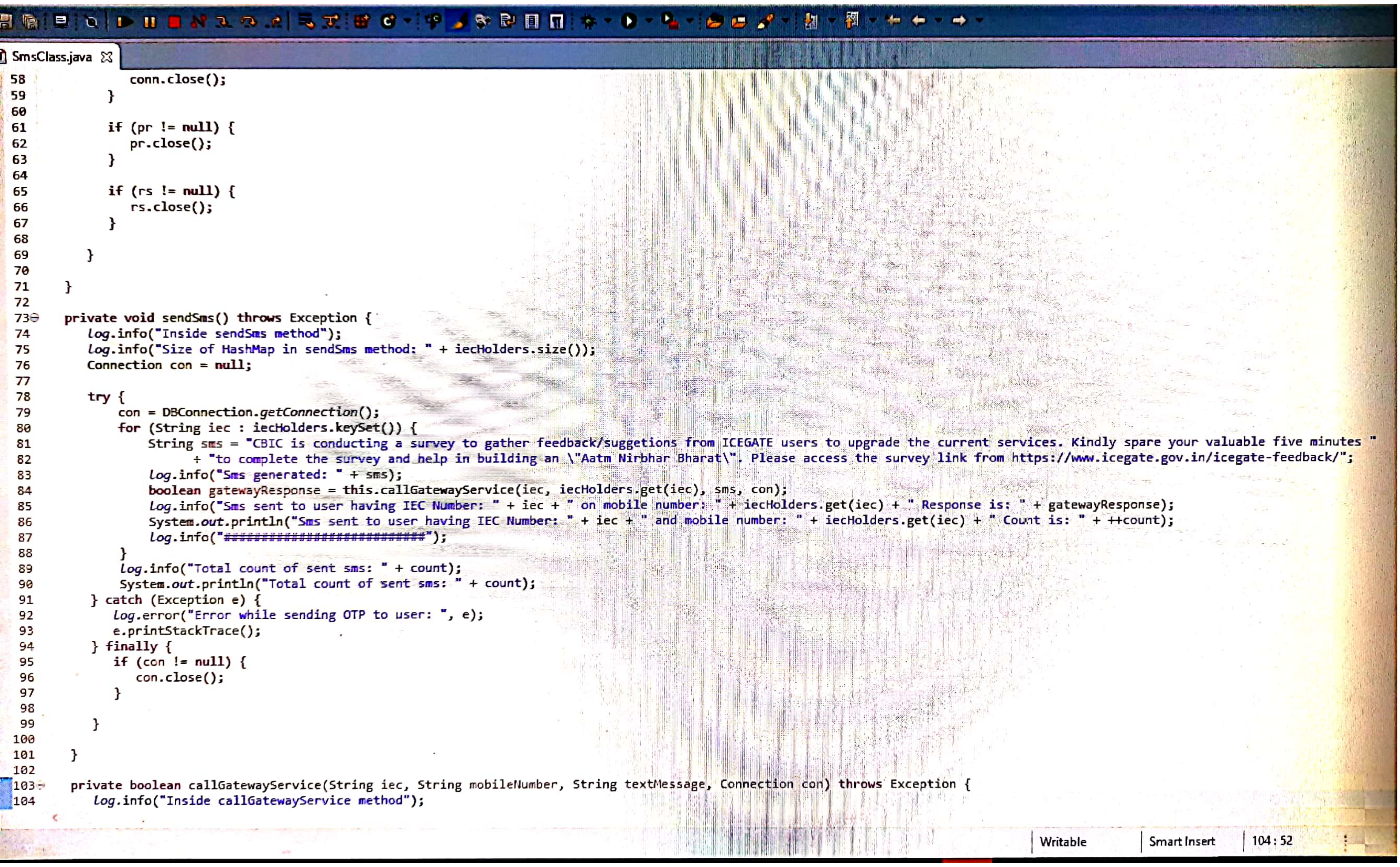
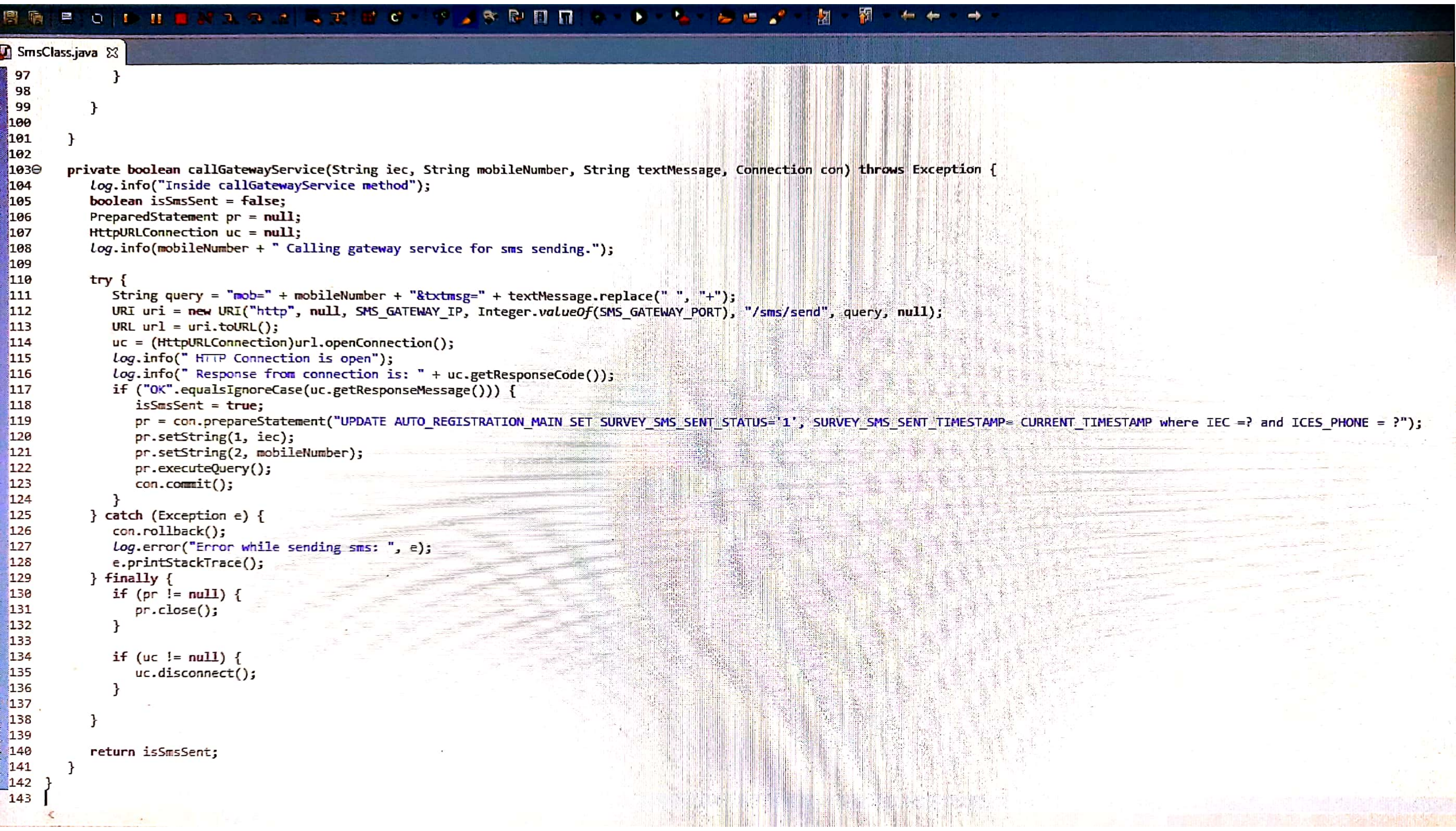


```java
package in.gov.icegate.controller;

import in.gov.icegate.common.db.DBConnection;□

public class SmsClass {
   private static Logger log = Logger.getLogger(SmsClass.class);
   private Map<String, String> iecHolders = null;
   private int count = 0;
   private String SMS_GATEWAY_IP = "10.1.2.156";
   private String SMS_GATEWAY_PORT = "8080";

   public static void main(String[] args) throws Exception {
      log.info("Inside main method of SurveySMSUtility");
      SmsClass smsClass = new SmsClass();
      smsClass.getData();
      smsClass.sendSms();
      log.info("End of main method of SurveySMSUtility");
      log.info("########################################################################################################################");
   }

   private void getData() throws Exception {
      log.info("Inside getData method");
      Connection conn = null;
      PreparedStatement pr = null;
      ResultSet rs = null;
      iecHolders = new HashMap<String, String>();

      try {
         System.out.println("Inside getData");
         conn = DBConnection.getConnection();
         pr = conn.prepareStatement("SELECT IEC,ICES_PHONE FROM AUTO_REGISTRATION_MAIN where (SURVEY_SMS_SENT_STATUS is NULL or SURVEY_SMS_SENT_STATUS != '1') and ROWNUM<=1000");
         rs = pr.executeQuery();

         while(rs.next()) {
            if (rs.getString(1) != null && rs.getString(2) != null && rs.getString(1) != "" && rs.getString(2) != "") {
               iecHolders.put(rs.getString(1), rs.getString(2));
            }
         }

         System.out.println("Size of HashMap is: " + iecHolders.size());
         log.info("Size of HashMap is: " + iecHolders.size());
      } catch (Exception e) {
         log.error("Datababse error: ", e);
         e.printStackTrace();
      } finally {
         if (conn != null) {
            conn.close();
```



```java
                conn.close();
            }

            if (pr != null) {
                pr.close();
            }

            if (rs != null) {
                rs.close();
            }

        }

    }

    private void sendSms() throws Exception {
        log.info("Inside sendSms method");
        log.info("Size of HashMap in sendSms method: " + iecHolders.size());
        Connection con = null;

        try {
            con = DBConnection.getConnection();
            for (String iec : iecHolders.keySet()) {
                String sms = "CBIC is conducting a survey to gather feedback/suggetions from ICEGATE users to upgrade the current services. Kindly spare your valuable five minutes "
                        + "to complete the survey and help in building an \"Aatm Nirbhar Bharat\". Please access the survey link from https://www.icegate.gov.in/icegate-feedback/";
                log.info("Sms generated: " + sms);
                boolean gatewayResponse = this.callGatewayService(iec, iecHolders.get(iec), sms, con);
                log.info("Sms sent to user having IEC Number: " + iec + " on mobile number: " + iecHolders.get(iec) + " Response is: " + gatewayResponse);
                System.out.println("Sms sent to user having IEC Number: " + iec + " and mobile number: " + iecHolders.get(iec) + " Count is: " + ++count);
                log.info("##########################");
            }
            log.info("Total count of sent sms: " + count);
            System.out.println("Total count of sent sms: " + count);
        } catch (Exception e) {
            log.error("Error while sending OTP to user: ", e);
            e.printStackTrace();
        } finally {
            if (con != null) {
                con.close();
            }

        }

    }

    private boolean callGatewayService(String iec, String mobileNumber, String textMessage, Connection con) throws Exception {
        log.info("Inside callGatewayService method");
```



```
            }

        }

    }

    private boolean callGatewayService(String iec, String mobileNumber, String textMessage, Connection con) throws Exception {
        log.info("Inside callGatewayService method");
        boolean isSmsSent = false;
        PreparedStatement pr = null;
        HttpURLConnection uc = null;
        log.info(mobileNumber + " Calling gateway service for sms sending.");

        try {
            String query = "mob=" + mobileNumber + "&txtmsg=" + textMessage.replace(" ", "+");
            URI uri = new URI("http", null, SMS_GATEWAY_IP, Integer.valueOf(SMS_GATEWAY_PORT), "/sms/send", query, null);
            URL url = uri.toURL();
            uc = (HttpURLConnection)url.openConnection();
            log.info(" HTTP Connection is open");
            log.info(" Response from connection is: " + uc.getResponseCode());
            if ("OK".equalsIgnoreCase(uc.getResponseMessage())) {
                isSmsSent = true;
                pr = con.prepareStatement("UPDATE AUTO_REGISTRATION_MAIN SET SURVEY_SMS_SENT_STATUS='1', SURVEY_SMS_SENT_TIMESTAMP= CURRENT_TIMESTAMP where IEC =? and ICES_PHONE = ?");
                pr.setString(1, iec);
                pr.setString(2, mobileNumber);
                pr.executeQuery();
                con.commit();
            }
        } catch (Exception e) {
            con.rollback();
            log.error("Error while sending sms: ", e);
            e.printStackTrace();
        } finally {
            if (pr != null) {
                pr.close();
            }

            if (uc != null) {
                uc.disconnect();
            }

        }

        return isSmsSent;
    }
}
```