uat_adhoc_invalid.pdf	
Invalid Adhoc Ticket	Date: 22/09/2017

Test Name	Invalid Adhoc Ticket
Use Case Tested:	Pay for Ticket
Test Description:	A customer passes a casual (a.k.a. 'Adhoc') ticket to the paystation, etc. however the barcode is invalid.
Pre-conditions	The car park is open for business and someone intends to pay for their casual carpark ticket.
Post-conditions	An invalid ticket message has been displayed.
Notes:	**The paystation ui is launched from the IDE.**
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	EXPECTED TEST RESULTS	P	F
1.	An 'issued' ticket that is not an 'in use' (a.k.a. 'current') ticket is inserted by driver.	System rejects the ticket.		**F**

uat_adhoc_invalid.pdf		
Invalid Adhoc Ticket	Date: 22/09/2017	

Test Item		Description	Result
1.	Ticket Number: 1 DateString:22092017060633 A-1- FEFF0032003200300039003200300031003700300036003000360033003300	This data was copied and then pasted into the ui barcode reader after the ticket was issued but the status was 'issued' not 'current'.	Fail

The paystation did not display a message "Go to the office" or reject the ticket. The code fails at:

Exception in thread "AWT-EventQueue-0" java.lang.ArithmeticException: Non-terminating decimal expansion; no exact representable decimal result.

       at java.math.BigDecimal.divide(BigDecimal.java:1690)

       at bcccp.carpark.CalcAdhocTicketCharge.calcDayCharge(CalcAdhocTicketCharge.java:252)
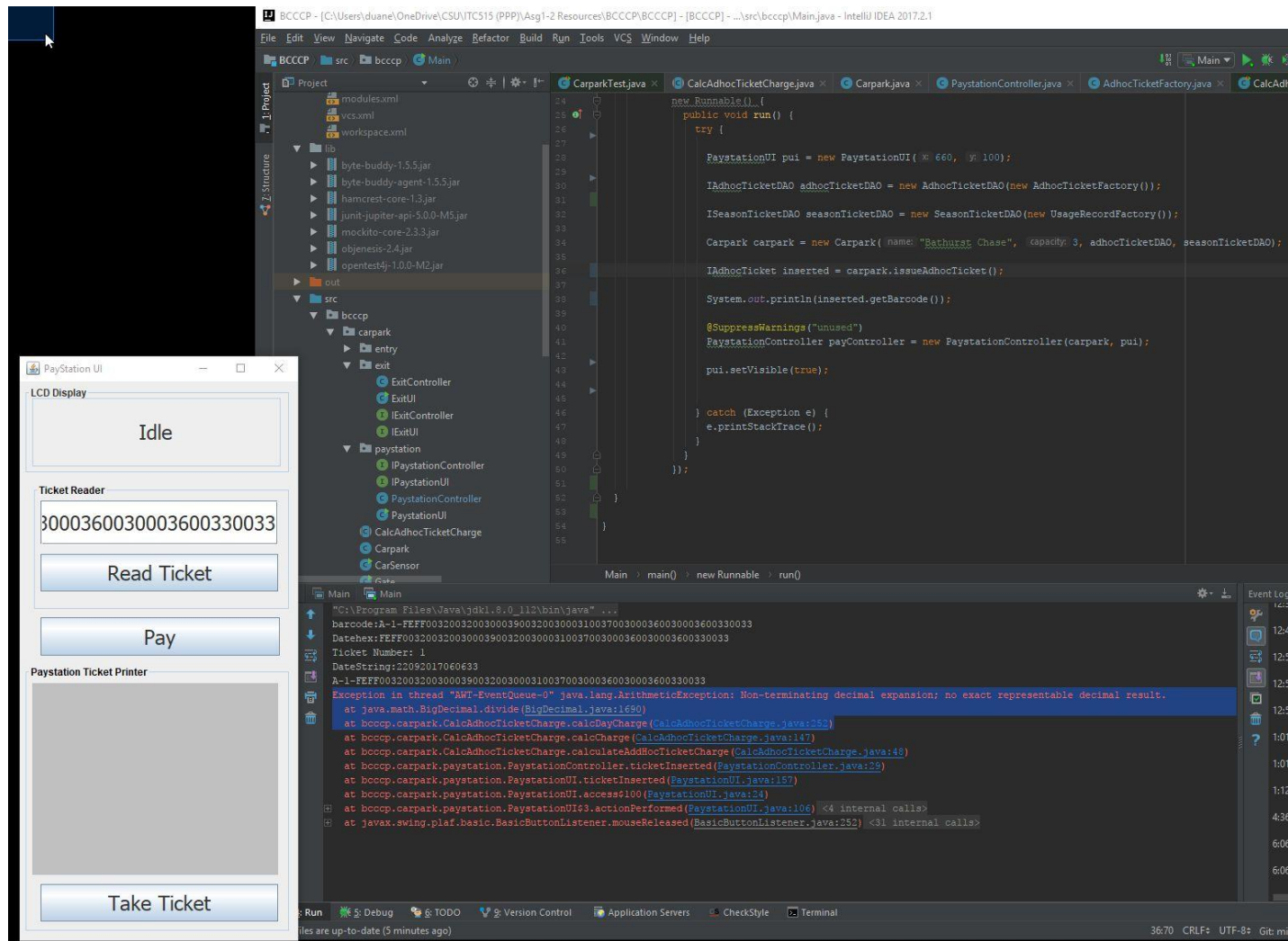
Figure 1A technically invalid ticket is submitted.

Script:

```java
package bcccp;


import bcccp.tickets.adhoc.IAdhocTicket;
import java.awt.EventQueue;
import bcccp.carpark.Carpark;
import bcccp.carpark.paystation.PaystationController;
import bcccp.carpark.paystation.PaystationUI;
import bcccp.tickets.adhoc.AdhocTicketFactory;
import bcccp.tickets.adhoc.AdhocTicketDAO;
import bcccp.tickets.adhoc.IAdhocTicketDAO;
import bcccp.tickets.season.ISeasonTicketDAO;
import bcccp.tickets.season.SeasonTicketDAO;
import bcccp.tickets.season.UsageRecordFactory;

import java.util.Date;


public class Main {

  public static void main(String[] args) {

    EventQueue.invokeLater(
            new Runnable() {
              public void run() {
                try {

                    PaystationUI pui = new PaystationUI(660, 100);

                    IAdhocTicketDAO adhocTicketDAO = new AdhocTicketDAO(new AdhocTicketFactory());

                    ISeasonTicketDAO seasonTicketDAO = new SeasonTicketDAO(new UsageRecordFactory());

                    Carpark carpark = new Carpark("Bathurst Chase", 3, adhocTicketDAO, seasonTicketDAO);

                    IAdhocTicket inserted = carpark.issueAdhocTicket();
```

```java
            System.out.println(inserted.getBarcode());

            @SuppressWarnings("unused")
            PaystationController payController = new PaystationController(carpark, pui);

            pui.setVisible(true);


        } catch (Exception e) {
            e.printStackTrace();
        }
      }
    });

  }

}
```