1 INTRODUCTION
This lab can be divided into three main parts:
- Familiarization
- Manual functional testing
- Regression testing

This lab is a simple introduction to some of the concepts inherent to software testing. This includes hands on experience in testing an example software system, and following industrial defect tracking practices through several phases of defects' life cycles. The focus of this lab is to provide good issue reporting practice. Also students will be able to compare manual and regression testing strategies.

Sections 2.1 through 2.3 will be performed in pairs. The lab report will be completed and submitted by each student group. Make sure to include the names of both students in the lab report.

1.1 DUE DATE AND LATE MARKING POLICY
Lab report should be submitted through the submission form on the course home page (https://courses.cs.ut.ee/2017/SWT2017/spring/Main/Submission). Deadline for submissions is 7 days from the current lab session. For example if the lab has been done on a Tuesday, the deadline is on the following Monday at 23:59.

Late reports: There will be 50% penalty for 1 day delay. There will be no credit points for reports turned in after the 1 day delay date.

1.2 EXAMPLE REPOSITORIES OF REAL DEFECTS
The following are example repositories of real defects for real projects in the software industry. Make sure to review many examples to see how other experienced testers write defect reports.
- https://issues.apache.org/jira/browse/FLEX/?selectedTab=com.atlassian.jira.jira-projects-plugin:issues-panel

An example with explanations can be found in Appendix A (the screen shot has been taken from Bugzilla bug tracking tool).
2 Instructions

The system under test for this lab is an ATM simulation system. To use the ATM simulation system simply download "ATM System - Lab 1 Version 1.0.jar" from the course wiki page, save it in a known location and run that file from the saved location. The system should begin execution with the GUI as shown in Appendix B.

There are two valid hard-coded card numbers and PINs:

Card Number: 1 PIN: 42 Available Accounts: Checking and Savings
Card Number: 2 PIN: 1234 Available Accounts: Checking and Money Market

Note: Both of these cards access the same checking account.

The initial balances are:

Checking: $100 Savings: $1000 Money Market: $5000

2.1 Familiarization with the System Under Test

1. If you haven’t done so already, download the ATM simulation system version 1.0 from the course wiki page.
2. Run the JAR file version 1.0 file to show the GUI as shown in Appendix B.

Perform a deposit

3. Turn the system on using the “On” button.
4. Enter the number of $20 bills that the system is assumed to start off with, noting that this is the number of bills, not the total value of the bills. Entering a value of 10 for instance indicates that the ATM is starting with $200 (10 twenty dollar bills). Any number greater than 0 will suffice for now.
5. Click on the “Click to insert card” button which has now appeared on the main interface below the simulated ATM display.
6. The screen now changes to a prompt for the user to input the card number (since there is no actual physical card reader). Enter 1 for the card number and press Enter. Upon returning to the main screen, the display is now requesting the PIN be entered.
7. Type 42 using the simulated keypad and press Enter. The display now prompts the user to perform one of four transactions: withdraw, deposit, balance inquiry, or transfer.
8. Press 2 on the simulated keypad to perform a deposit. The display now prompts the user to indicate which account they would like to deposit to: checking, savings, or the money market account.
9. Press 2 on the simulated keypad to deposit to the savings account. The display now prompts the user to enter the deposit amount.
10. Enter any positive amount and press Enter. A button which represents the user inserting the deposit envelope now appears.
11. Click that button to simulate inserting the envelope. The display now prompts the user whether they wish to perform another transaction or not.
12. Press 2 on the simulated keypad to indicate you do not wish to perform another transaction. The main window shows a button appearing, simulating the ejecting of the user’s card.
13. Press the System Power Button once again to turn the ATM system off.
2.2 **FAMILIARIZATION WITH TARGETPROCESS**

2.2.1 **Logging in**

Log in to the TargetProcess account that has been created for you. You should have received an email with a sign-in link. The password of the first log in will be used for future log ins.

If you didn't receive an e-mail about account creation or you are not sure which e-mail address has been used, please ask the TA for assistance. Follow the link in your e-mail notification. You should now be logged in and viewing the TargetProcess main page.

On first log in you will be presented with a welcoming video which you can skip. After the video there is also a welcoming tour. You are recommended to also skip the tour from top right corner of the screen.

2.2.2 **Reporting a defect**

Begin by navigating to the bugs view, by clicking on the “Bugs” button on the left side of the main page.

To add a new defect report click on the “Add” button at the top left corner of the screen. Then select the “Bug” option, add a summary and click on the “Add & Open” button. A screenshot of the form is shown below:

![Bug Report Form](image)

Next you will be able to see a detailed view of the report. In this view you are required to fill out the description field according to the example in Appendix A. You should also set the current state of the bug and the version number to the build being tested. This additional fields are located on the right side of the
form. If the needed build number is missing, then create it analogically to creating a bug using the “Add” button in top left corner of the main page. A screenshot of setting the build version is shown below.

Test out creating a bug report now, before proceeding forward. Think of a defect that you have seen (or make one up) in your least favorite piece of software, and enter details in the Summary and Description fields. When filling in the description, try to include high level description of the defect, the steps to reproduce, the actual results (what happened), and the expected results (what should have happened).

2.2.3 Displaying all defects for a project

To display all the defects navigate to the bugs view, by clicking on the “Bugs” button on the left side of the main page. The list can also be filtered, by clicking on the filter field at top of the bugs view.

2.3 Manual Functional Testing

This section is to be performed in pairs. One student can ‘drive’ the testing (operate the computer executing the system under test), while the other student keeps track of which tests have been performed, reports any defects found, and determines what order to execute tests in.

14. In Appendix D, a basic test suite has been provided for this SUT. You can pick arbitrary number of test cases from the test suite. Execute selected test cases at least once, verifying that the actual results match the expected results for each case. Report any defects found as described below. We expect you to find and report at least 9 defects.

15. Reporting defects: While performing your testing, make sure to record the following information for each test case into description field:

- The function being tested (e.g., Login)
- The initial state of the system (e.g., System is on and is idle, i.e., not already serving a customer)
- The action to take (e.g., Insert a card, enter correct card number and PIN)
- The expected outcome (e.g., the system successfully accepts the customer, and shows the banking menu)
• While performing the tests, if any of the actual results differ from the expected results, report that as a defect in the report.

16. Reporting defects in a simple, concise manner. When reporting defects, follow the guidelines in the Sample Bug Report article (http://www.softwaretestinghelp.com/sample-bug-report), and use the same style as in Appendix A.

17. When reporting these defects, ensure that version 1.0 is selected in the “Build” field.

2.4 DEFECT CORRECTION VERIFICATION AND REGRESSION TESTING

This section is to be performed in pairs. The defects reported in the two previous stages of testing can be divided among the group and can be retested individually.

18. Download the updated version (version 1.1) of the ATM simulation system from course page. This version of the system has been partially fixed by imaginary developers based on the defect reports previously existing.

19. Retest each of these defects to determine which have been fixed and which have not. To do this we must follow the defect lifecycle as shown in Appendix C. Since we do not know which defects have been fixed exactly, assume that all defects have had an attempt to fix them (to emulate the developer's progress on the bug open the issue by changing the state to “In Progress” and then to “Resolved”. If the defect has actually been fixed in the ATM system version 1.1, change the status to “Verified”. If the defect has not actually been fixed in the ATM system version 1.1, change the status to “Reopened” and write a comment stating “Defect still exists in version 1.1”.

20. When reporting these defects, ensure that version 1.1 is selected.

2.5 SUMMARY

Within your group, you should now each have a general understanding of how to effectively report and track defects. You have also progressed through a short iteration of manual functional testing and an iteration of regression testing.

3 DELIVERABLES AND GRADING

Both students of each group must attend the lab session. The maximum number of points is 10. Points are allocated as follows:

3.1 IN-LAB DEMONSTRATION OF PROGRESS (1 POINT)

You will be required to demonstrate your progress to the TA during lab time.

Note: If you work in a pair, both students of each group should be present in the demo times.

3.2 TARGETPROCESS DEFECT REPORTS (6 POINTS)

Students will be graded on their defect reports AS SAVED IN THE TARGETPROCESS SYSTEM. Please do not repeat your bug reports in your lab reports! The grading criteria (rubric) for defect reports are as follows.
### TargetProcess Defect Report (6 POINTS)

1. Correctness: Do the defect reports contain the detailed defect information? Does the report contain all the defects in the same level of detail? Does it contain the input, the expected output, and the faulty output for each defect?  
   - 3 points

2. Clarity and adherence to defect reporting guidelines: Is it obvious where to start (what state to bring the program to) to replicate the defect? Is it obvious what you would type? Is it obvious what files to use (if any)?  
   - 3 points

Important: we will not count points for the defects reported in TargetProcess if the lab report in pdf format is not submitted through the course wiki page.

### 3.3 Lab Report (3 points)

**Important:** lab report should be submitted through the submission form on the course home page. LAB REPORT SUBMISSION IS TREATED AS A TIMESTAMP OF FINISHED WORK. IF STUDENTS DO NOT SUBMIT THE REPORT WE ASSUME THAT THE WORK IS NOT DONE. Lab reports should be submitted in PDF format only. Deadline for submissions is 7 days from current lab session.

Students will be required to submit one report per group. In the report should be included:

<table>
<thead>
<tr>
<th>Lab Report (3 POINTS)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Number of defects found. Outline defects and corresponding test cases (it’s enough to put a reference to the TargetProcess bug report). Also mark whether defect has been fixed in version 1.1 or not. We expect to see at least 10 bug reports.</td>
<td>2 points</td>
</tr>
<tr>
<td>4. General comments and conclusions on performing the lab. Did you find it a useful practice? Was it easy to follow? Did you spend too little/too much time on it? Should some parts be dropped or explained better? Is something missing? Etc. Please try to be constructive.</td>
<td>1 point</td>
</tr>
</tbody>
</table>

### 4 Interesting Quotes and Web Links

- “Bug Advocacy: Lesson 59 - Take the time to make your bug reports valuable”
- “Bug Advocacy: Lesson 57 - Make your bug report an effective sales tool”
- “Bug Advocacy: Lesson 62 - Report perceived quality gaps as bugs”
- “Bug Advocacy: Lesson 67 - Report defects promptly”
- “Bug Advocacy: Lesson 68 - Never assume that an obvious bug has already been filed”
- “Bug Advocacy: Lesson 72 - Minor bugs are worth reporting and fixing”
- “Bug Advocacy: Lesson 84 - Never exaggerate your bugs”
- “Bug Advocacy: Lesson 85 - Report the problem clearly, but don’t try to solve it”

From Cem Kaner’s book “Lessons Learned in Software Testing”

5 ACKNOWLEDGEMENTS

The original version of this lab package was developed by Prof. Vahid Garousi at the University of Calgary and is part of a large software testing laboratory courseware available under a Creative Commons license for other testing educators.

We are also grateful for improvement suggestions proposed and introduced into these lab instructions by Oliver Vilson of Celeg Hannas OÜ and by Svetlana Omelkova.

6 REFERENCES


APPENDIX A B U G REPORT EXAMPLE

27.3.2014

Bug 1466

Summary: Quick Quote - Retrieving Postcode works only once after page load
Product: [Blackout] Reporter: Petteri Lyttinen
Component: General Assignee: Oliver Vilson
Status: NEW --- Severity: normal CC:
Priority: PS
Version: pre-live
Hardware: PC
OS: Windows

<table>
<thead>
<tr>
<th>Time tracking:</th>
<th>Orig. Est.</th>
<th>Actual Hours</th>
<th>Hours Worked</th>
<th>Hours Left</th>
<th>%Complete</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Deadline:

Petteri Lyttinen 2013-05-07 15:21:21 EEST

Description

Retrieving Postcode under "Vehicle kept address" only works once after page load. If the address lightbox is closed, it can not be opened up again by clicking the "Retrieve Address" button until the page has been reloaded. Modifying the postcode has no effect.

How to reproduce:
1. Navigate to the [Blackout] demo site
2. Start filling a new Quick Quote
3. Enter a postcode under the "Vehicle kept address" and click "Retrieve Address" (or TAB out of the postcode field)
4. Don't select an address from the lightbox that should have opened up but click "Close" instead
5. Click "Retrieve address" again

Expected result:
Address lightbox opens up again and user can select an address.

Actual result:
Nothing happens when user clicks "Retrieve Address" if the lightbox has already been closed once eg. due to user typing the postcode and going back to fix it.

Tested on 64-bit Windows 7 with:
- Firefox 20.0.1
- Internet Explorer 10.0
- Google Chrome 24.0
- Safari 5.1.7
- Opera 12.15

Petteri Lyttinen 2013-05-07 15:26:15 EEST

Comment 1

Naturally, manually refreshing the page to get the address retrieval working again clears all data from the fields (except for name and date of birth, for some reason).

Petteri Lyttinen 2013-05-10 18:17:45 EEST

Comment 2

Postcode searching is restricted to avoid repeated requests to the postcode lookup service for the same postcode as there's a per lookup cost involved.

Agree if user doesn't select an address it should allow manual entry or same lookup again.

Currently implementing manual address lookup in the rare case that the postcode doesn't find the user's address so will pick this up as part of that.

http://dev.channels.eso.org/bugzilla/show_bug.cgi?id=1466

Page 8 of 14
APPENDIX B – USER INTERFACE FOR ATM SIMULATION SYSTEM

- Simulated ATM Display
- Simulated Receipt Printer
- Simulated Keypad
- Access to the System Log
- System Power Button
APPENDIX C – DEFECT LIFECYCLE FROM INITIAL REPORT TO CLOSED

- Create
  - OPEN
    - IN PROGRESS
      - RESOLVED
      - VERIFIED
    - REOPENED
  - CLOSED
### APPENDIX D - FUNCTIONAL TEST SUITE

<table>
<thead>
<tr>
<th>Test Case #</th>
<th>Use Case</th>
<th>Function Being Tested</th>
<th>Initial System State</th>
<th>Input</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Startup</td>
<td>System is started when the switch is turned &quot;on&quot;</td>
<td>System is off</td>
<td>Activate the &quot;on&quot; switch</td>
<td>System requests initial cash amount</td>
</tr>
<tr>
<td>2</td>
<td>System Startup</td>
<td>System accepts initial cash amount</td>
<td>System is requesting cash amount</td>
<td>Enter a legitimate amount</td>
<td>System is on</td>
</tr>
<tr>
<td>3</td>
<td>System Startup</td>
<td>Connection to the bank is established</td>
<td>System has just been turned on</td>
<td>Perform a legitimate inquiry transaction</td>
<td>System output should demonstrate that a connection has been established to the Bank</td>
</tr>
<tr>
<td>4</td>
<td>System Shutdown</td>
<td>System is shut down when the switch is turned &quot;off&quot;</td>
<td>System is on and not servicing a customer</td>
<td>Activate the &quot;off&quot; switch</td>
<td>System is off</td>
</tr>
<tr>
<td>5</td>
<td>Session</td>
<td>System reads a customer's ATM card</td>
<td>System is on and not servicing a customer</td>
<td>Insert a readable card</td>
<td>Card is accepted; System asks for entry of PIN</td>
</tr>
<tr>
<td>6</td>
<td>Session</td>
<td>System rejects an unreadable card</td>
<td>System is on and not servicing a customer</td>
<td>Insert an unreadable card</td>
<td>Card is ejected; System displays an error screen; System is ready to start a new session</td>
</tr>
<tr>
<td>7</td>
<td>Session</td>
<td>System accepts customer's PIN</td>
<td>System is asking for entry of PIN</td>
<td>Enter a PIN</td>
<td>System displays a menu of transaction types</td>
</tr>
<tr>
<td>8</td>
<td>Session</td>
<td>System allows customer to perform a transaction</td>
<td>System is displaying menu of transaction types</td>
<td>Perform a transaction</td>
<td>System asks whether customer wants another transaction</td>
</tr>
<tr>
<td>9</td>
<td>Session</td>
<td>System allows multiple transactions in one session</td>
<td>System is asking whether customer wants another transaction</td>
<td>Answer yes</td>
<td>System displays a menu of transaction types</td>
</tr>
<tr>
<td>10</td>
<td>Session</td>
<td>Session ends when customer chooses not to do another transaction</td>
<td>System is asking whether customer wants another transaction</td>
<td>Answer no</td>
<td>System ejects card and is ready to start a new session</td>
</tr>
<tr>
<td>11</td>
<td>Transaction</td>
<td>System handles an invalid PIN properly</td>
<td>A readable card has been entered</td>
<td>Enter an incorrect PIN and then attempt a transaction</td>
<td>The Invalid PIN Extension is performed</td>
</tr>
<tr>
<td>12</td>
<td>Withdrawal</td>
<td>System asks customer to choose an account to withdraw from</td>
<td>Menu of transaction types is being displayed</td>
<td>Choose Withdrawal transaction</td>
<td>System displays a menu of account types</td>
</tr>
<tr>
<td>13</td>
<td>Withdrawal</td>
<td>System asks</td>
<td>Menu of account</td>
<td>Choose</td>
<td>System displays a menu of</td>
</tr>
<tr>
<td>Test Case #</td>
<td>Use Case</td>
<td>Function Being Tested</td>
<td>Initial System State</td>
<td>Input</td>
<td>Expected Output</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>-----------------------</td>
<td>----------------------</td>
<td>-------</td>
<td>-----------------</td>
</tr>
<tr>
<td>14</td>
<td>Withdrawal</td>
<td>System performs a legitimate withdrawal transaction properly</td>
<td>types is being displayed</td>
<td>checking account</td>
<td>possible withdrawal amounts</td>
</tr>
<tr>
<td>15</td>
<td>Withdrawal</td>
<td>System verifies that it has sufficient cash on hand to fulfill the request</td>
<td>System has been started up with less than the maximum withdrawal amount in cash on hand; System is requesting a withdrawal amount</td>
<td>Choose an amount that the system currently has and which is not greater than the account balance</td>
<td>System displays an appropriate message and asks customer to choose a different amount</td>
</tr>
<tr>
<td>16</td>
<td>Withdrawal</td>
<td>System verifies that customer's balance is sufficient to fulfill the request</td>
<td>System is requesting a withdrawal amount</td>
<td>Choose an amount that the system currently has but which is greater than the account balance</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>17</td>
<td>Withdrawal</td>
<td>A withdrawal transaction can be cancelled by the customer any time prior to choosing the dollar amount</td>
<td>System is displaying menu of account types</td>
<td>Press 'Cancel' key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>18</td>
<td>Withdrawal</td>
<td>A withdrawal transaction can be cancelled by the customer any time prior to choosing the dollar amount</td>
<td>System is displaying menu of dollar amounts</td>
<td>Press 'Cancel' key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>19</td>
<td>Deposit</td>
<td>System asks customer to choose an account to deposit to</td>
<td>Menu of transaction types is being displayed</td>
<td>Choose Deposit transaction</td>
<td>System displays a menu of account types</td>
</tr>
<tr>
<td>20</td>
<td>Deposit</td>
<td>System asks customer to enter a dollar amount to deposit</td>
<td>Menu of account types is being displayed</td>
<td>Choose checking account</td>
<td>System displays a request for the customer to type a dollar amount</td>
</tr>
<tr>
<td>21</td>
<td>Deposit</td>
<td>System asks customer to insert an envelope</td>
<td>System is displaying a request for the customer to type a legitimate</td>
<td>Enter a legitimate</td>
<td>System requests that customer insert an envelope</td>
</tr>
<tr>
<td>Test Case #</td>
<td>Use Case</td>
<td>Function Being Tested</td>
<td>Initial System State</td>
<td>Input</td>
<td>Expected Output</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>------------------------------------------------------------</td>
<td>-------------------------------------------</td>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>22</td>
<td>Deposit</td>
<td>System performs a legitimate deposit transaction properly</td>
<td>System is requesting that customer insert an envelope</td>
<td>Insert an envelope</td>
<td>System accepts envelope; System prints a correct receipt showing amount and correct updated balance; System records transaction correctly in the log (showing message to the bank, approval back, and acceptance of the envelope)</td>
</tr>
<tr>
<td>23</td>
<td>Deposit</td>
<td>A deposit transaction can be cancelled by the customer any time prior to inserting an envelope</td>
<td>System is displaying menu of account types</td>
<td>Press 'Cancel' key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>24</td>
<td>Deposit</td>
<td>A deposit transaction can be cancelled by the customer any time prior to inserting an envelope</td>
<td>System is requesting customer to enter a dollar amount</td>
<td>Press 'Cancel' key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>25</td>
<td>Deposit</td>
<td>A deposit transaction can be cancelled by the customer any time prior to inserting an envelope</td>
<td>System is requesting customer to insert an envelope</td>
<td>Press 'Cancel' key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>26</td>
<td>Transfer</td>
<td>System asks customer to choose an account to transfer from</td>
<td>Menu of transaction types is being displayed</td>
<td>Choose Transfer transaction</td>
<td>System displays a menu of account types specifying transfer from</td>
</tr>
<tr>
<td>27</td>
<td>Transfer</td>
<td>System asks customer to choose an account to transfer to</td>
<td>Menu of account types to transfer from is being displayed</td>
<td>Choose checking account</td>
<td>System displays a menu of account types specifying transfer to</td>
</tr>
<tr>
<td>28</td>
<td>Transfer</td>
<td>System asks customer to enter a dollar amount to transfer</td>
<td>Menu of account types to transfer to is being displayed</td>
<td>Choose savings account</td>
<td>System displays a request for the customer to type a dollar amount</td>
</tr>
<tr>
<td>29</td>
<td>Transfer</td>
<td>System performs a legitimate transfer transaction properly</td>
<td>System is displaying a request for the customer to type a dollar amount</td>
<td>Enter a legitimate dollar amount</td>
<td>System prints a correct receipt showing amount and correct updated balance; System records transaction correctly in the log (showing both message to the bank and approval back)</td>
</tr>
<tr>
<td>30</td>
<td>Transfer</td>
<td>A transfer transaction can be cancelled by the customer any time prior to entering</td>
<td>System is displaying menu of account types specifying transfer from</td>
<td>Press 'Cancel' key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>Test Case #</td>
<td>Use Case</td>
<td>Function Being Tested</td>
<td>Initial System State</td>
<td>Input</td>
<td>Expected Output</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>-----------------------</td>
<td>----------------------</td>
<td>-------</td>
<td>-----------------</td>
</tr>
<tr>
<td>31</td>
<td>Transfer</td>
<td>A transfer transaction can be cancelled by the customer any time prior to entering dollar amount</td>
<td>System is displaying menu of account types specifying transfer to</td>
<td>Press &quot;Cancel&quot; key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>32</td>
<td>Transfer</td>
<td>A transfer transaction can be cancelled by the customer any time prior to entering dollar amount</td>
<td>System is requesting customer to enter a dollar amount</td>
<td>Press &quot;Cancel&quot; key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
<tr>
<td>33</td>
<td>Inquiry</td>
<td>System asks customer to choose an account to inquire about</td>
<td>Menu of transaction types is being displayed</td>
<td>Choose Inquiry transaction</td>
<td>System displays a menu of account types</td>
</tr>
<tr>
<td>34</td>
<td>Inquiry</td>
<td>System performs a legitimate inquiry transaction properly</td>
<td>System is displaying menu of account types</td>
<td>Choose checking account</td>
<td>System prints a correct receipt showing correct balance; System records transaction correctly in the log (showing both message to the bank and approval back)</td>
</tr>
<tr>
<td>35</td>
<td>Inquiry</td>
<td>An inquiry transaction can be cancelled any time prior to choosing an account</td>
<td>System is displaying menu of account types</td>
<td>Press &quot;Cancel&quot; key</td>
<td>System displays an appropriate message and offers customer the option of choosing to do another transaction or not.</td>
</tr>
</tbody>
</table>