

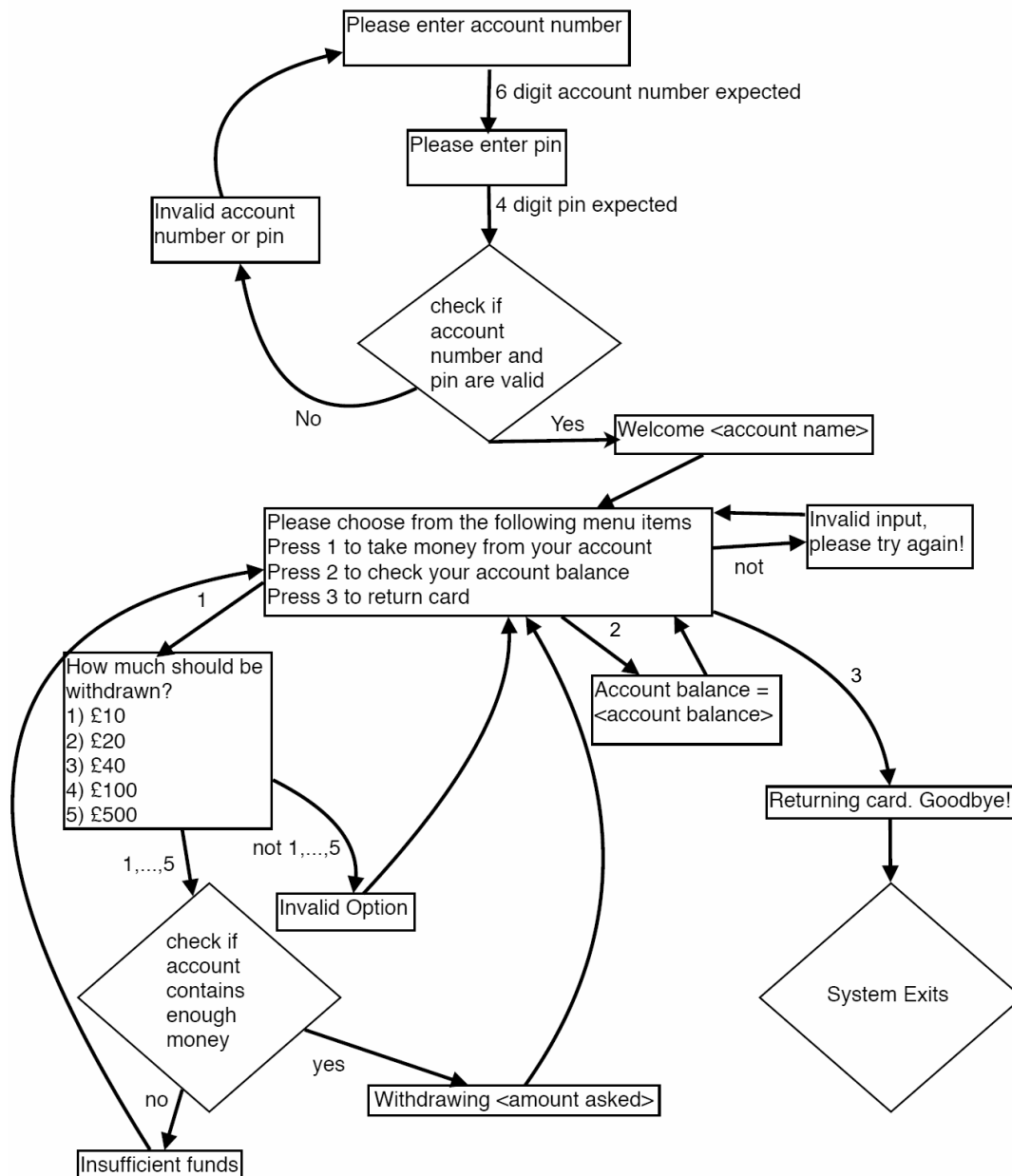
AC22005 - Computer Systems 2B

Coursework 3 - ATM Simulator using Threads in C#

This development project is designed to allow you to demonstrate your understanding of threads and semaphores in C#.

Automated Teller Machines

A bank has many Automated Teller Machines, and any bank account may have more than one ATM card, hence more than one user may attempt to access the same account at the same time - this could cause a concurrency problem if the account balance is being modified by two transactions simultaneously. This problem can be avoided by implementing an appropriate semaphore/locking scheme for each ATM to use. The overall system looks like this:



Requirements

Your task is to create a multiple-window C# application in Visual Studio 2010, which shows the bank system and a number of ATMs. Suitable interface features should be included to simulate normal ATM functions, and the *stages of a transaction should be able to be stepped through manually so that you can demonstrate different transactions going through together*.

Some sample C# code is given - your first task is to familiarize yourself with this code:

- Compile the code, run it and try all the different options. After this, change the operation so that rather than exiting when a person takes their card (option 3), the system goes back to the beginning.
- Modify the code so that **more than one ATM runs at once**, from this central computer. *Hint*: this should be done by making each ATM run in its own thread.
- Modify your code so that two people share the same account number and balance; it should be possible to **demonstrate inconsistency** (caused by a data race) between the transactions made and changes to the account balance if transactions are performed "simultaneously".
- You should then modify the system to include an appropriate mechanism to make sure that you do **not have a data race** occurring that would mean that the same account balance is not being modified at the same time, and hence be able to demonstrate that the same inconsistency *cannot* now occur; this mechanism should be focussed **on the critical code only**, hence *maximising concurrency* of other stages/features while still *preventing inconsistency*.

Submission & Assessment

This coursework is due for submission via MyDundee (under AC22005 Assignments) at 5pm on **Friday 21st March 2014** (Friday of Week 10) and is worth 13% of your total grade for this module. You should submit a ZIP file containing your **whole VS 2010 C# project folder** (including *all* source files and executable) and a **written report** of 400-600 words (include a word count) describing features of your program, your approach to the problem, any difficulties you encountered and any features that you have added. Keep a copy of what you submit in case there are problems with your submission.

During the lab class in week 11, you will make a short presentation of your ATM system to Dr Murray - this is your opportunity to draw our attention to the best parts of your solution. Your demonstration should include demonstration of the inconsistent system and then of the fixed (consistent) system. The demonstrations will constitute a part of the marking process for this coursework so attendance is compulsory, and you should check in advance that your solution functions correctly on the lab computers with Visual Studio 2010.

Marking Guidance

A basic ATM simulation will achieve 40 marks (out of 80). A fully-functioning system which demonstrates both inconsistent and consistent behaviour will achieve 58 marks out of 80. Marks will be lost for poor or incomplete functionality or coding, or for a poorly constructed or unappealing user interface. Additional marks will be given for a particularly engaging user interface, sophisticated interactions or other appropriate extra features. A well-written report which meets all the given guidelines, with a good demonstration will achieve 15 marks (out of 20). A simple solution with good report will thus achieve 55% (C2) while a complete solution with good report will thus achieve 73% (A3).