AC22005

ryan crampton, rOSS mAIDER

Computer Systems 2B:ATM

2

Contents

[The approach to the project 2](#_Toc4508746)

[Features to the ATM 2](#_Toc4508747)

[Difficulties encountered 2](#_Toc4508748)

[Solutions 2](#_Toc4508749)

[Conclusion 3](#_Toc4508750)

## The approach to the project

The initial step we assessed was which code we had and what we had to create, upon analysing the code we had, we realised the core to the project was the GUI changeover from console and the threading itself, to this extent we could divide the tasks among us, with one handling the initial transfer of the code, followed by both working on sections of the GUI and coding itself. Once it was on a GUI, Methods were the easiest way to divide out the tasks, we divided out the methods as the one who handled the port to console had the methods such as checking balance, while the transferring of money was left to the other

Overall the approach was made simple through communication and working together to divide up tasks.

## Features to the ATM

The main features the ATM has is that upon start-up, you can decide if you would like to run the ATM with or without race conditions, once you have decided that, this then also allows you to select with or without Semaphore testing, before proceeding to the login page

From the login page, you can enter the account you would like to enter and select the “open an ATM” button, this opens the account page for the user, this is possible to have multiple of, and the system expects at least 2 when opened. This will open two separate ATMs with the account specified. When you are in the ATM itself, you can choose to withdraw cash, check balance and exit, if you select check balance, you get a info bar explaining the balance in the account. When you withdraw/choose to withdraw the money options come up, this allows you to select which amount you wish to withdraw, once you have selected this, the system will wait for the other terminal to select, then depending on the conditions, update the balance accurately or having met a race condition only display once removed. At every stage, it is possible to exit the system. This is normally with an Exit button.

## Difficulties encountered

The main difficulty encountered is that when trying to create the race condition, it was a lot harder than making it work as an ATM should, we spent a lot of time trying to figure out how this would work, and even made attempts at fixing another solution around it, none of this worked upon initial testing and we couldn’t break what was working for a while.

Another problem that we encountered was using the barriers within threading, this links to problem one as it allowed the system to pull stuff at the same time, however it wasn’t acting as intended initially and permanently blocked a channel if it was not working and clearing channels wasn’t working it.

## Solutions

The solution to the initial problem was to move how the account interacted with the program, when the programme was moved over and we considered how this would interact with the system itself, it made the solution clearer as if you had a variable that it updated itself then changed on system after that, it would cause an inconsistency.

The second solution was solved through the barrier in that the signal and wait, this allowed the system to wait on all ATMs to be ready before it would allow them through, this solved the issue

## Conclusion

Overall the project went smoothly and was solved with clear clarity, in future development the project would go a lot smoother as the skills picked up could be carried over.