OOP Assignment 1c Report

Question 1

a)

According to the UML Class Diagram in Fig. 2 is it possible for the canWithdraw

operation to be inherited and used directly in the ChildAccount class? Describe what

problems may arise when implementing this design and what can be done to solve

these.

b)

The C++ ATM class includes a BankAccount pointer as one of its data members. Explain

why, in this case study, using a BankAccount instance instead of a pointer to

implement this relationship, would not be appropriate. Give specific examples from

your final solution to illustrate your answer.

c)

In the C++ implementation given, what is the nature of relationship between the Card

and List<string> classes, how should it represented in UML and what C++

mechanisms are involved in its C++ implementation?

d)

Is the UserInterface class an abstract class? How do you know? If not, should it be?

e)

Why is the Date::currentDate() function declared as static? How does this

mechanism work?

f)  
Assuming that t1 is a valid Time instance, indicate which functions are called in each of

the following lines of code. Will they work with the Time class given? Explain the issues,

if any, and describe what changes are needed in the Time class for each of these

statements to be valid, equivalent and works as expected (i.e., create the same instance

t).

Time t(t1 + Time(0, 0, 12)); //line 1

Time t(Time(12) + t1); //line 2

Time t(t1 + 12); //line 3

Time t(12 + t1); //line 4

g)

The purpose of the TransactionList::deleteGivenTransaction function is to

delete the first occurrence, if there is one, of a given transaction from a non-empty

transaction list. Is the following version correct? If not, describe all the reasons why it

isn’t and show how it should be amended to work correctly.

void TransactionList::deleteGivenTransaction(const Transaction& tr)

{

assert(size() != 0);

if (newestTransaction() == tr)

\*this = olderTransactions();

else

{

Transaction firstTr(newestTransaction());

olderTransactions().deleteGivenTransaction(tr);

this->addNewTransaction(firstTr);

}

}

h)

What differences would it make to declare a method such as

BankAccount::prepareFormattedAccountDetails as virtual? Explain why you

might want to do this.

i)

Would the expression p\_theActiveAccount\_->getOverdraftLimit() be valid if the

pointer p\_theActiveAccount\_ were to currently points to a CurrentAccount

instance? If not, explain what could be done, if anything, to solve this problem.

j)

Could the function ATM::m\_acct1\_produceBalance have equally been rewritten as

follow? If not, give all the reasons why not.

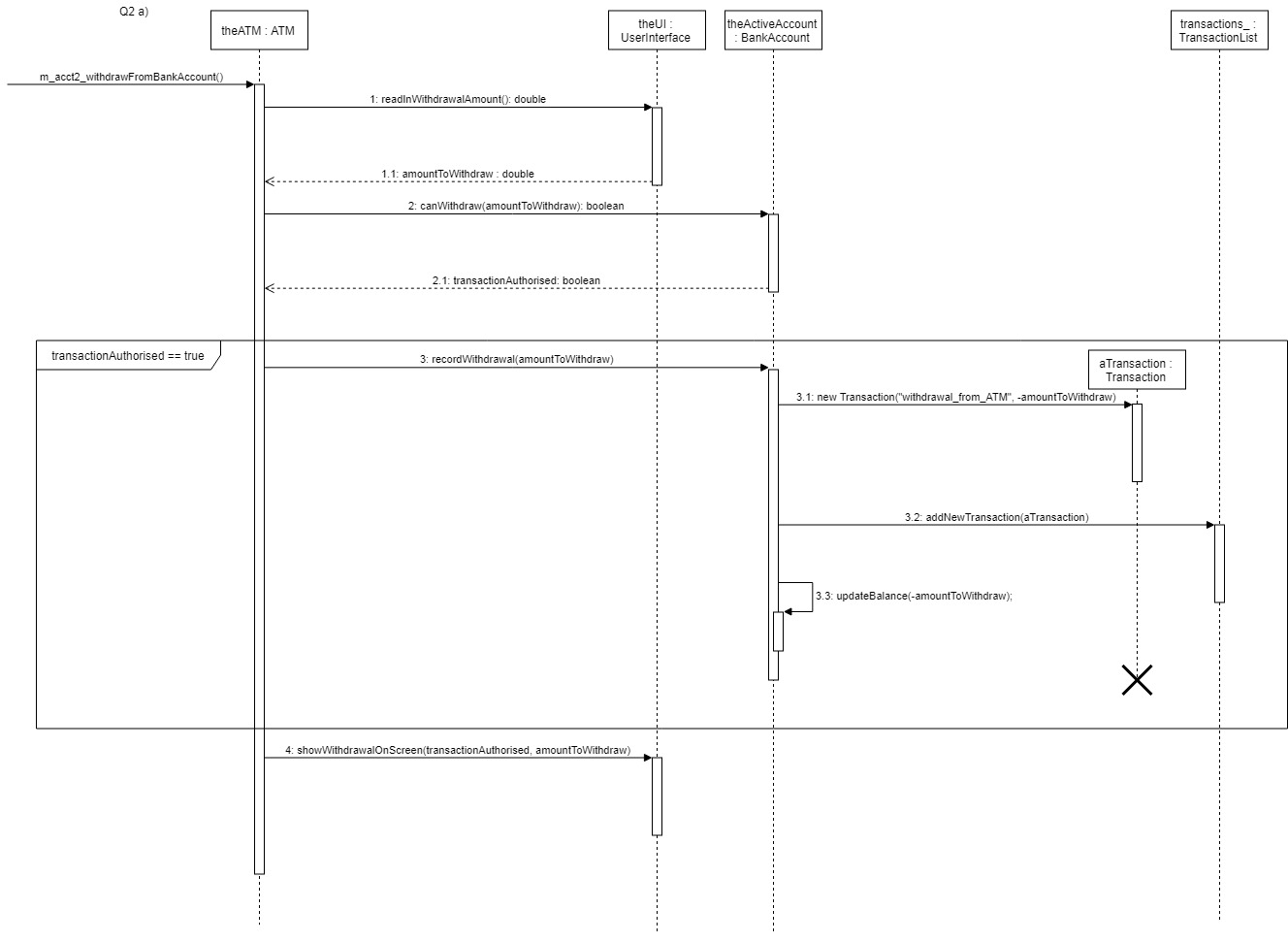
void ATM::m\_acct1\_produceBalance() const {

theUI\_.showProduceBalanceOnScreen(p\_theActiveAccount\_->balance\_);

}

Question 2

a)



b)

Question 3

Question 4

Question 5

Version control log

use git log > log.txt or git log | clip and paste here.

----------------------

Notes:

delete notes and question text after answering

spec:

1. A Word document for your group report containing your answers to Question 1, the UML diagrams for Question 2, C++ code extracts for Questions 3, 4 and 5 plus a version control log.

Answer questions about code provided: You must justify your answers to get the marks on this question.

UML diagrams: You must adopt the diagram conventions presented in class or specified in this document. However, if necessary, you can use English to complement your explanations so long as they are clear and unambiguous. You can produce the diagrams by hand or use tools such as Visio, Rational Rose or Together to help you.

C++ code: You must include the code for the required header files and specified function definitions.

Version Control Log: This report, automatically generated by the tool you have used to support the version control for the group work, lists the changes made by the various members of your team. It will be used to evidence individual contributions to the group work. So when make sure that the log information documents actual members' contributions.

Report: This report is to be presented on A4 paper typed, stapled (easily accessible for adding comments), paginated and identified with your assignment group number (e.g. SE1, CS4G3) and names of group members clearly indicated.