1. (10 pts) A requirements specification of the software that needs to be created in your own words. This should be a short English narrative describing what you think the software should do and what it will operated like, not how it will implement its functions.

The software will keep track of users at the bank. along with the user / customer, it will keep track of two accounts checking and savings. it will keep track of withdraws and deposits along with interest. In regard to the users it will organize them by age address name cell phone number and id.  based off the problem description of how it would function it sounds similar to an online banking app.

2. (10 pts) Detailed use cases for all the scenarios you imagine the software will be used. Look at the phone directory example in the text book (section 1.5) to guide you with this.

user insets command to deposit

system prompts for name

user inputs name

system prompts for id

user inputs id

system ask for checking (display amount) or savings (display amount)

user inputs savings

system displays deposit screen

user inputs amounts to deposit

system asks for conformation

users inputs ‘yes’

system adds transaction to transaction array

system adds amount to savings account

system displays new savings amount

program ends

user insets command to deposit

system prompts for name

user inputs name

system prompts for id

user inputs id

system ask for checking (display amount) or savings (display amount)

user inputs checking

system displays deposit screen

user inputs amounts to deposit

system asks for conformation

users inputs ‘yes’

system adds transaction to transaction array

system adds amount to checkings account

system displays new checking amount

program ends

user insets command to withdrawn

system prompts for name

user inputs name

system prompts for id

user inputs id

system ask for checking (display amount) or savings (display amount)

user inputs checking

system displays withdraw screen

user inputs amounts to withdrawn

system asks for conformation

users inputs ‘yes’

system adds transaction to transaction array

system subtracts amount from checking’s account

system displays new checking amount

system gives amount to user

program ends

user insets command to withdrawn

system prompts for name

user inputs name

system prompts for id

user inputs id

system ask for checking (display amount) or savings (display amount)

user inputs savings

system displays withdraw screen

user inputs amounts to withdrawn

system asks for conformation

users inputs ‘yes’

system adds transaction to transaction array

system subtracts amount from savings account

system displays new savings amount

system gives amount to user

program ends

user inserts command to add account

system asks for name

user inputs name

system ask for phone number

users inputs number

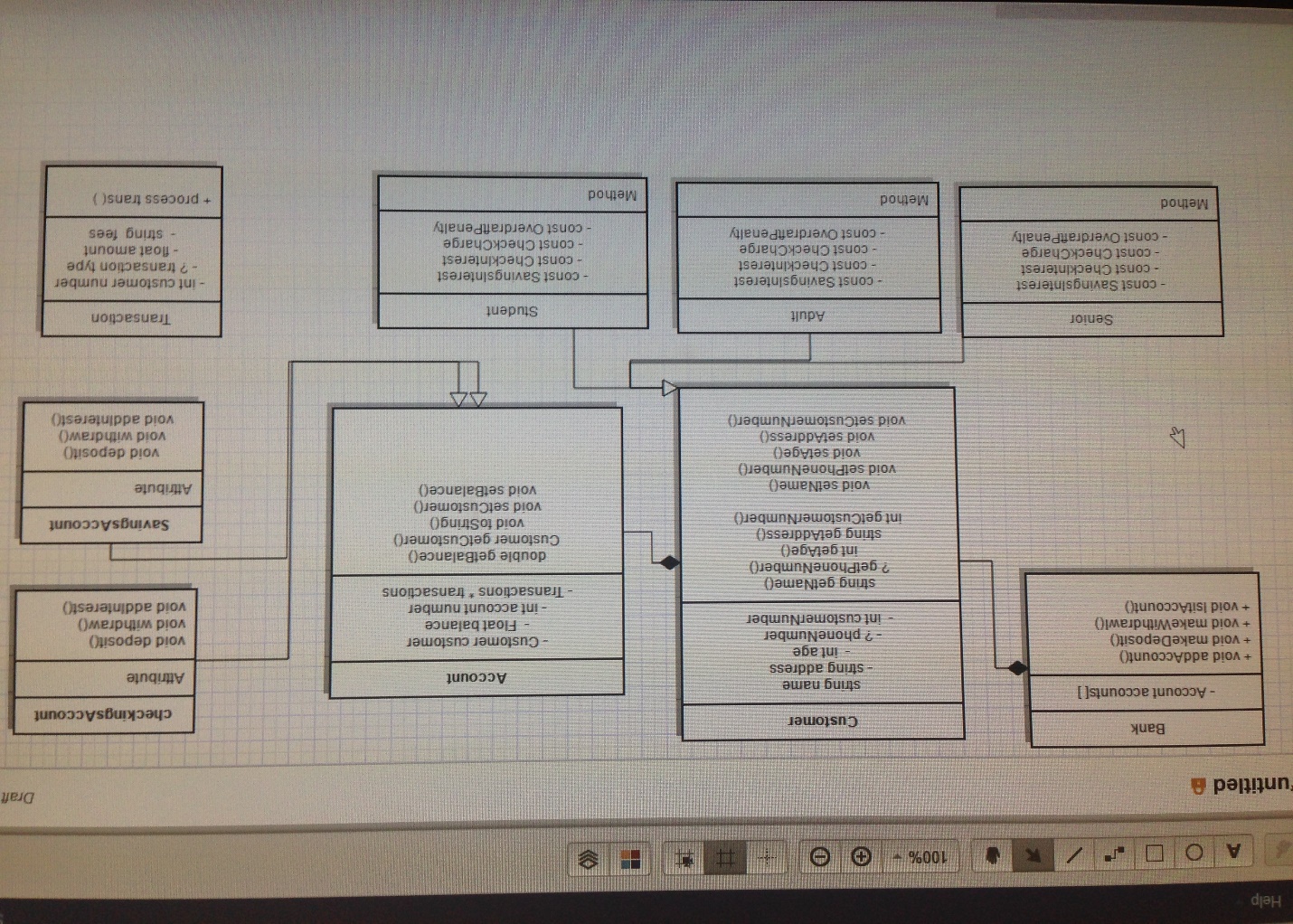
system ask for address

user inputs address

system displays users new account number

program ends

3. (10 pts) UML diagrams to describe the relationship between the classes described in the problem. You do not need to describe any sequence diagrams unless you wish too.



4. (10 pts) Pseudocode (see page 106 in your book for an example) to describe how the following methods will be implemented. Note, you will need to implement more code than is described here…

\* Add\_Account in Banking\_Application.cpp,

\* make\_deposit() methods in Bank.h and Banking\_Application.cpp,

\* make\_withdrawal() in Banking\_Application.cpp

\* Overloaded add\_account() methods in Bank.h

\* get\_account() in the Bank.h

5. (10 pts) Bank Data Storage Description – A description of HOW the account numbers for accounts and the customer id numbers will be generated and stored. How will accounts be linked to customers? How will transactions be linked to customers? (see Bank.h)

6 [123456]

7 [1234657]