**Architecture and Design**

**laboratory work 5**

**EXERCISE 1 – Actors, individuals, and the Banking Domain**

PURPOSE

The purpose of this exercise is to make concrete the difference between actors and individuals and to demonstrate why it is important to distinguish the way you think about them when developing software.

In this exercise, you will consider several kinds of actors and decide whether it matters if they are distinguished as individuals when designing the ATM (your project).

ACTIVITIES

Work as a team on this exercise. Perform each of the following activities. If you have questions, issues, or doubts, please ask for help and do not just guess.

1. Consider the following actors associated with a bank account:
   1. Account Owner
   2. Bank Teller
   3. Bank Manager

For each of these roles, consider the fallowing relationships with a bank account:

1. Account Owner deposits money
2. Account Owner withdraws money
3. Account Owner checks balance
4. Bank Teller checks balance
5. Bank Teller processes deposits
6. Bank Teller processes withdrawals
7. Account Manager opens accounts
8. Account Manager closes accounts
9. Account Manager waives fees
10. Account Manager authorizes overdraw payments

Given individuals named Paul and Maria, who are both qualified for any of these roles, discuss the following:

1. If Paul is both a teller and a customer of the bank, are there any potential conflicts?
   1. If yes, how should this conflict be reflected in the requirements for the software system?
2. If Maria is both a manager and a customer of the bank, are there any potential conflicts?
   1. If yes, how should this conflict be reflected in the requirements for the software system?
3. Describe two ways in which it would impact the bank’s software if it is possible for there to be more than one account manager per account. Also suggest ways in which the software could avoid/support the issues. Note that the impacts need not be negative.
4. If Maria and Paul are married and are both owners of an account, describe two ways in which this might impact the software requirements and suggest ways in which the software can deal with/support the impacts. Note that the impacts need not be negative.

**EXERCISE 2 – Creating Context Diagrams**

PURPOSE

This exercise gives your team the opportunity to think carefully about what the boundary of your ATM (project) software will be. You will not only decide what the limits of your software are but also how the software will interact with the elements in the environment with which it interacts, including people, hardware, and other software systems.

ACTIVITIES

1. Start by referring back to your team’s use case diagram. This will give you information about which elements in the environment will interact with your ATM (project) software.
2. Using one of the notations mentioned in class, create a top-level context diagram for your ATM’s C&C view. By this point you should have determined what kind of interaction mechanism will be used to interact with the external elements and whether they will be located on the same computer or over the Internet or intranet. Be as precise as you can be in describing the interactions.
3. Create a top-level context diagram for a module view.

**EXERCISE 18 – Major ATM Decisions**

PURPOSE

For this exercise, your team will consider any major decisions you made as to how the architecture will mitigate risks, including decisions that were made to ensure key quality concerns are addressed.

ACTIVITIES

1. Revisit your high-priority risks, consider whether any of them have been either mitigated or deemed to be less important.
2. For those that are architectural, record any major decisions that were made as to how the architecture will mitigate the risk. For each decision:
   1. State the decision
   2. Describe the risk that was addressed by the decision
   3. How the decision is reflected in the architecture
   4. Rejected alternatives (possible solutions that you did not go with)
   5. Why they were rejected
3. Revisit the project requirements focusing on the quality requirements.
4. Solutions to quality concerns such as security, safety, and performance are generally emergent – they involve multiple elements and relations of the system. Consider any major decisions that you have made related to your ATM (project) design that will help ensure quality goals are achieved. Record these as described in step 3 above. Later in the course, you will create views that reflect these decisions and will attach these decisions to the views.
5. When your team has completed the exercise, upload your document to the DL.