MacBank™ ABM

Presented by:
Yaseen Ali, Suraj Bhardwaj, Rohan Shah
Mechatronics Engineering

Group 302

Instructor: Dr. K. Sartipi

Course: SFWR ENG 3K04

Outline

- Overview
- SRS
- SRS to High-Level SDS
- High-Level SDS to Low-Level SDS
- Implementation in C#
- DEMO
- Conclusion

What is Software Engineering?

Application of:

- **Systematic**
- **Disciplined**
- **Quantifiable**
- Approach to the **development, operation** and **maintenance** of software.
- Study of these approaches is software engineering.

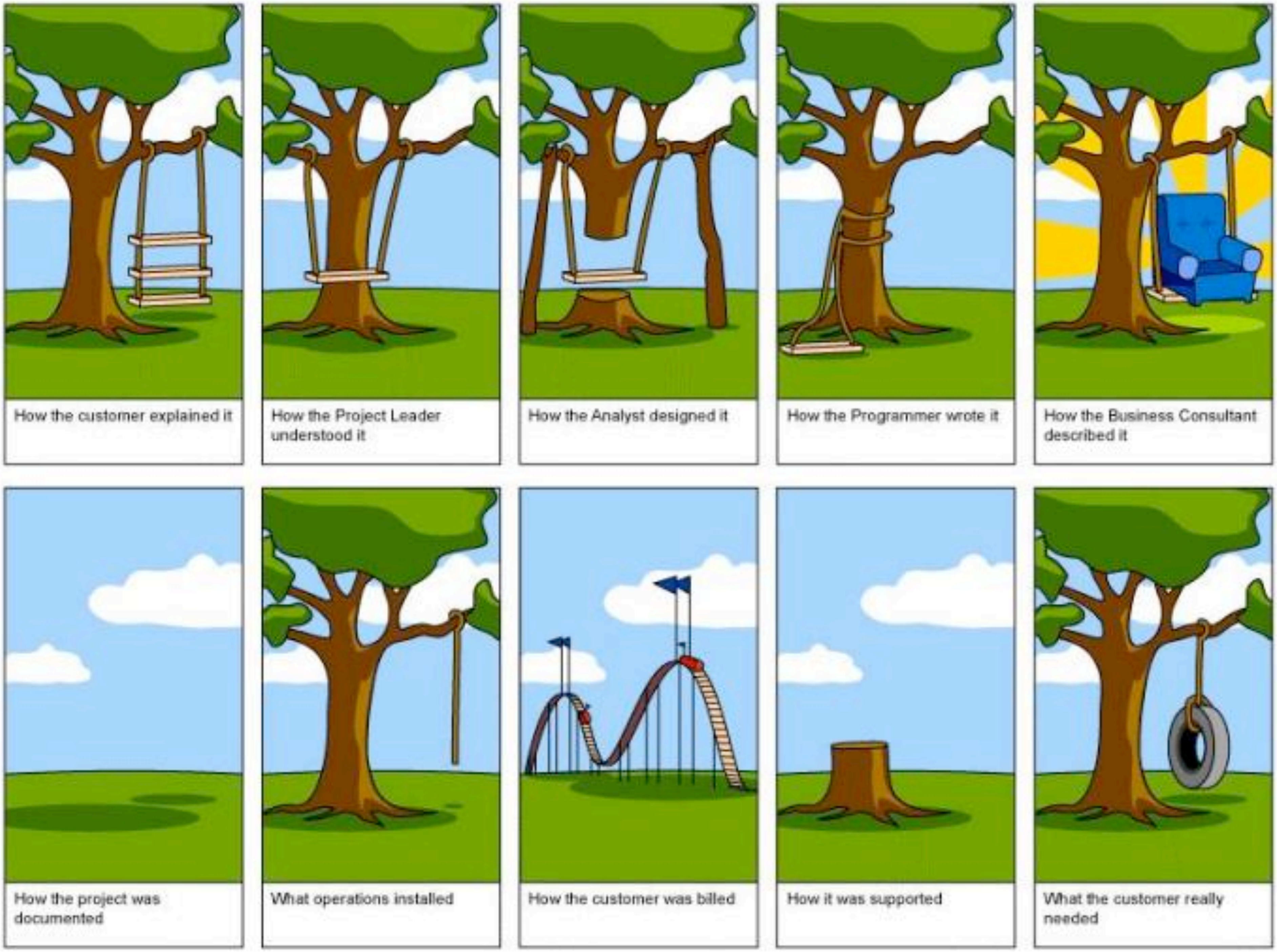
Software Engineering Crisis

- Current state of software technology in dealing with development and maintenance of large and complex software systems.

- Tackled by:

  - Providing disciplined processes and methodologies to:
    - Design
    - Implement
    - Maintain

  Large software systems.


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Software Engineering Project

- Consists of computer programs and associated documentation, which includes:
  - SRS
  - SDS
  - Technical Manual
  - User Manual
  - Source
  - Binary Files
  - Build Files
  - Version

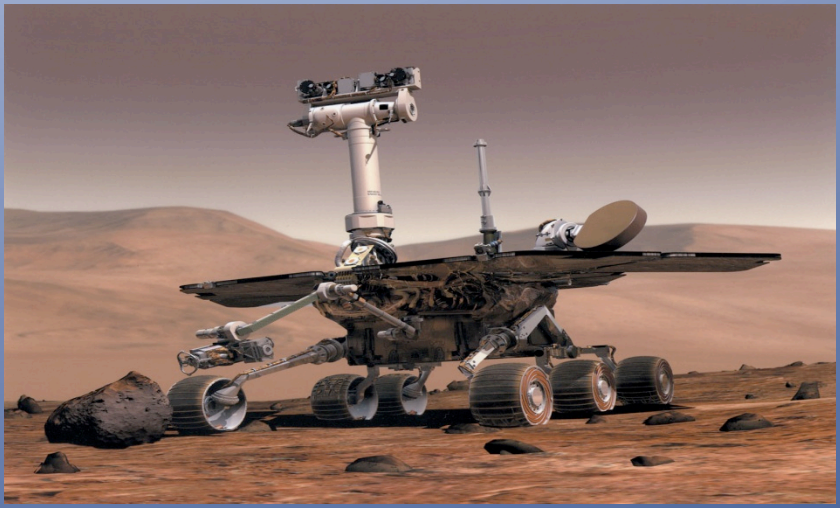
Why is it important?

Software engineering has had a major contribution in the development of the modern world. Software engineering is used in a variety of industries and applications:

- Process automation:

Why is it important?

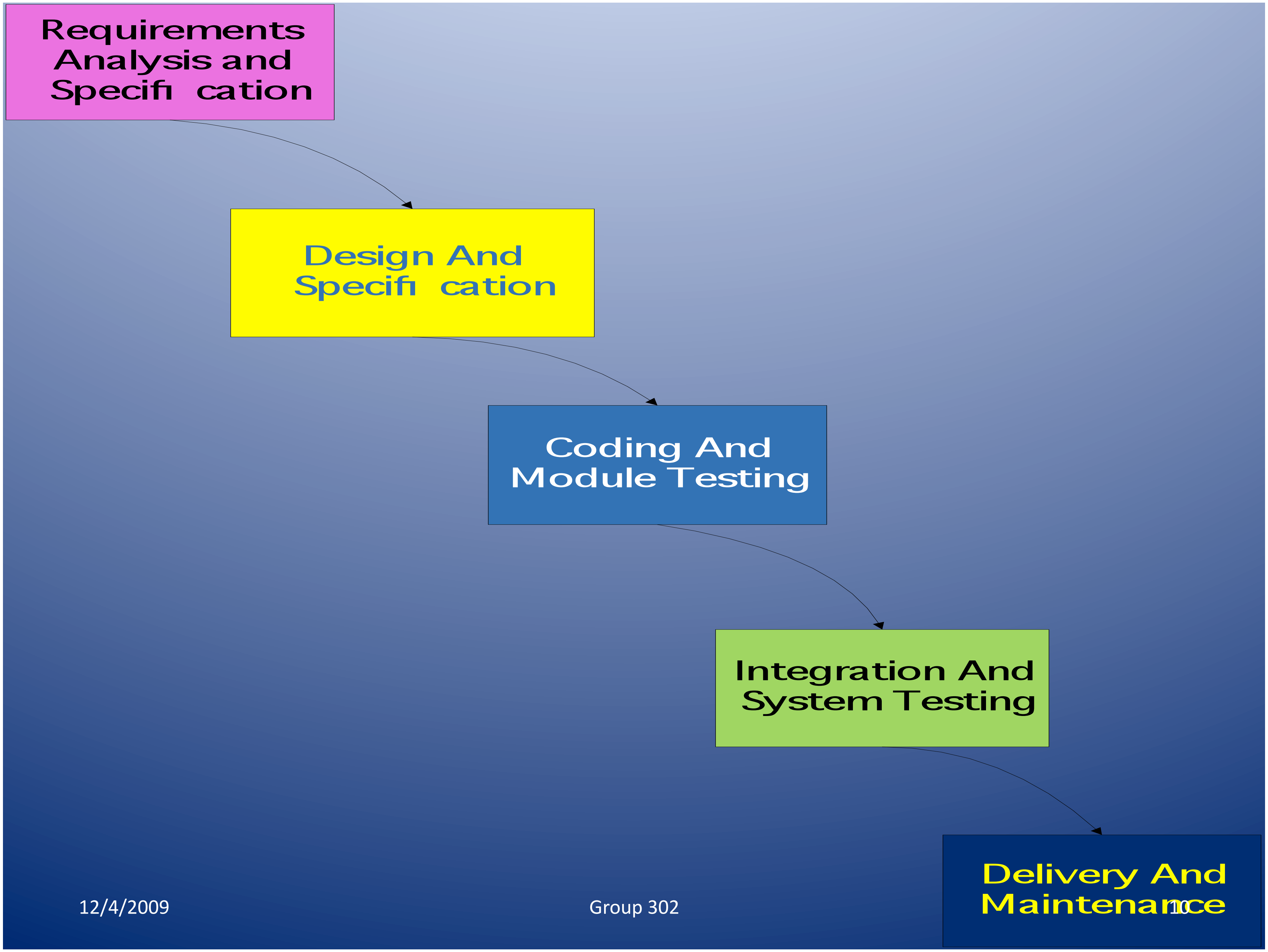
- Equipment control
- Scientific problems



- Entertainment industry
- Education and office management

Software Life Cycle Models

- Waterfall Model
- Extensive Programming
- Throwaway Prototyping Model
- Spiral Model
- Evolutionary Prototyping Model
- OSS Development Model


Requirements Analysis and Specifi cation
Design And Specifi cation
Coding And Module Testing
Integration And System Testing
Delivery And Maintenance

Requirements Analysis and Specifications

- After precise costs and benefits of the software have been defined.
- Requirements are in end user terms.
- Helps to make user manuals and system test plans.

Requirements and Specifications(ABM)

- 15 min interview with TA posing as MacBank representative.
- RFP provided
- Included basic banking requirements.
- Staff requirements.

SRS Benefits

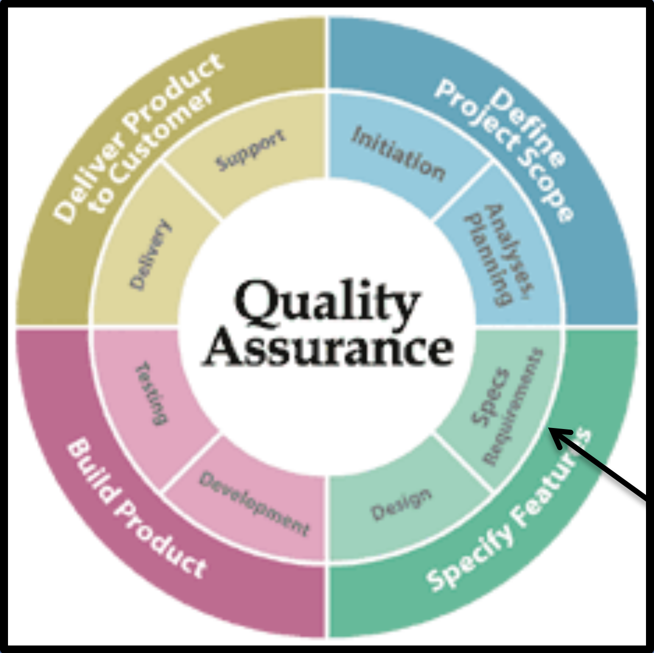
•SRS is the initial step in the software life cycle.

•SRS should contain everything there is to know about the system.

•Focuses on the “what” not the “how”

•It allows easy communication between individuals.

SRS and Quality



"The hardest single part of building a software system is deciding precisely what to build" (Frederick Brooks)

The Software Requirement Specification is an important step in providing a high quality software solution.

Features of SRS

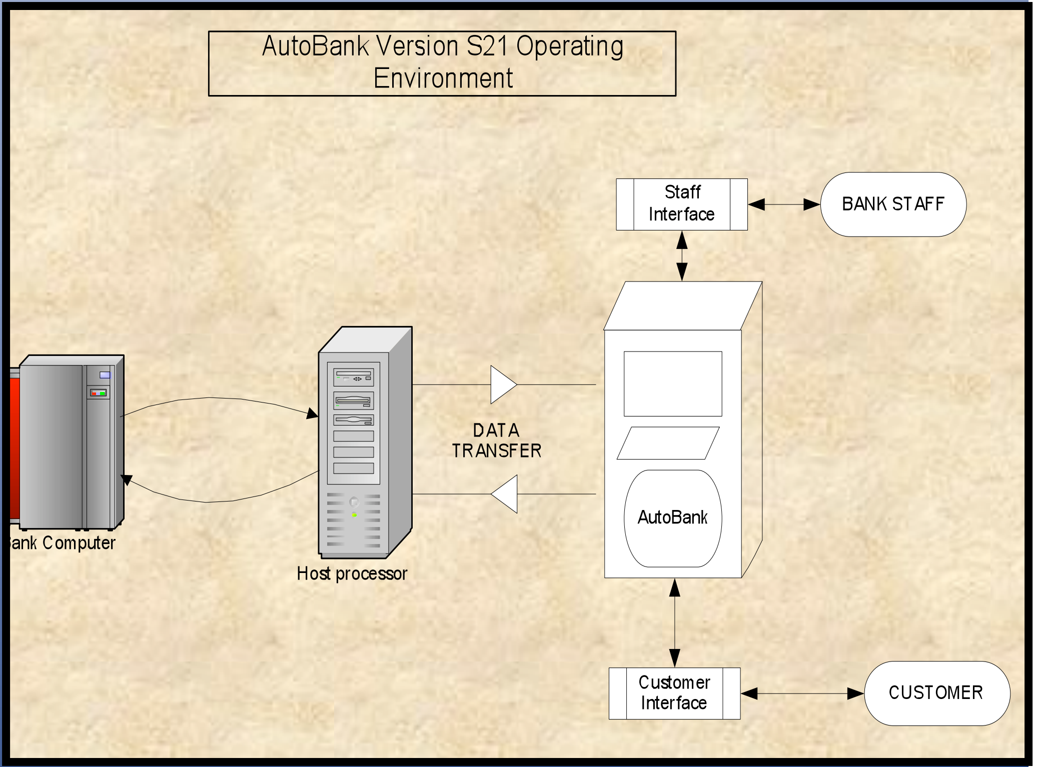
- Complete document specifying the goals of the system.
- Clearly defined what qualities the application must exhibit.
- Identified all stakeholders and defined the operating environment.
- Did NOT include how the solution was to be implemented.

SRS Environment

From the RFP :

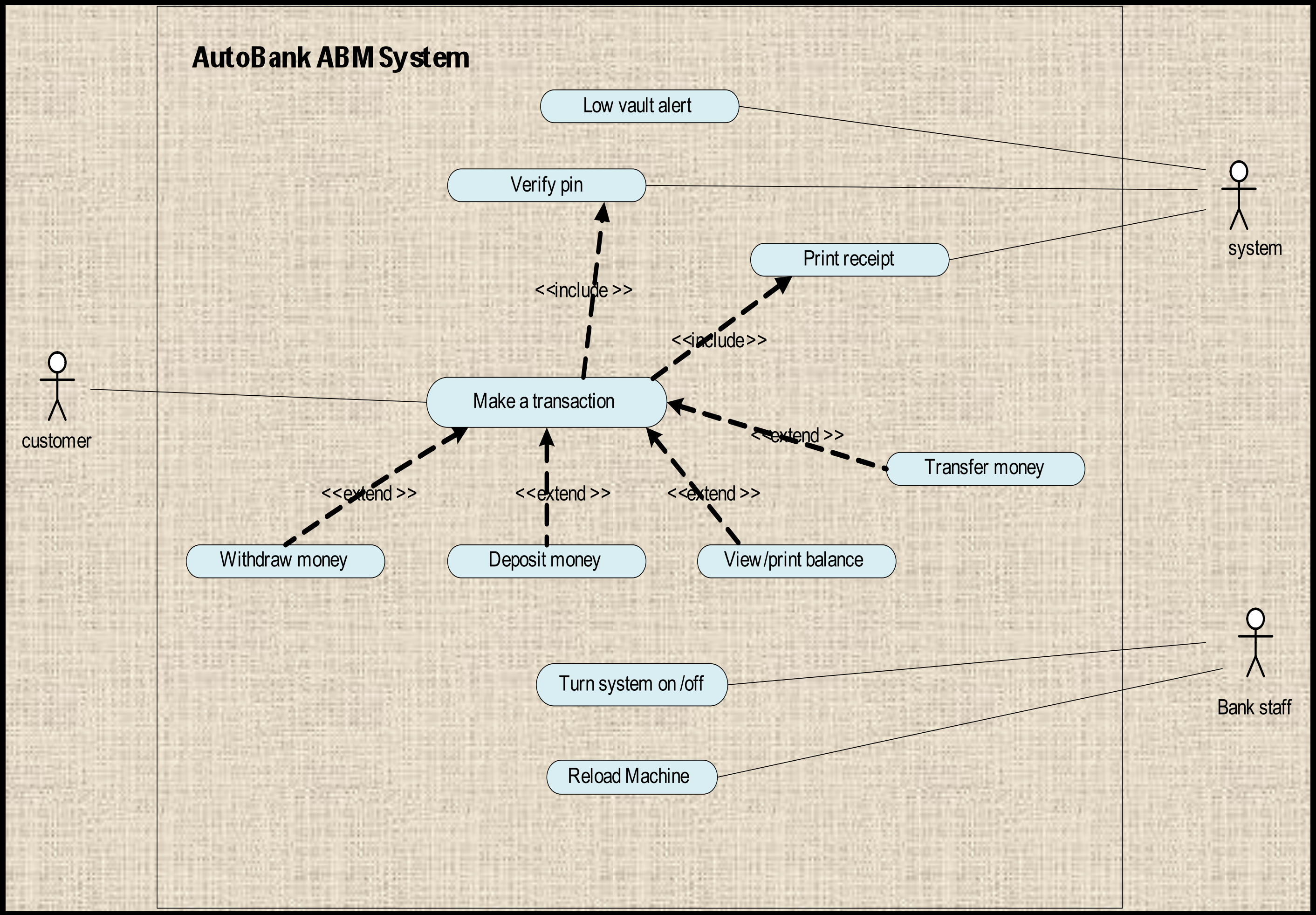
•Able to understand the operating environment.

•Translated requirements to Specifications document (SRS)



USE Case Diagram

Use Case diagram was used to specify the actors and actions in the system



Design and Specification

- 2 Sub-phases:
  - Architectural or High Level Design
  - Detailed or Low Level Design
- High Level: deals with overall module structure and organization
- Low Level: high level is then refined by
  - Designing each module in detail.
  - Solving individual problems

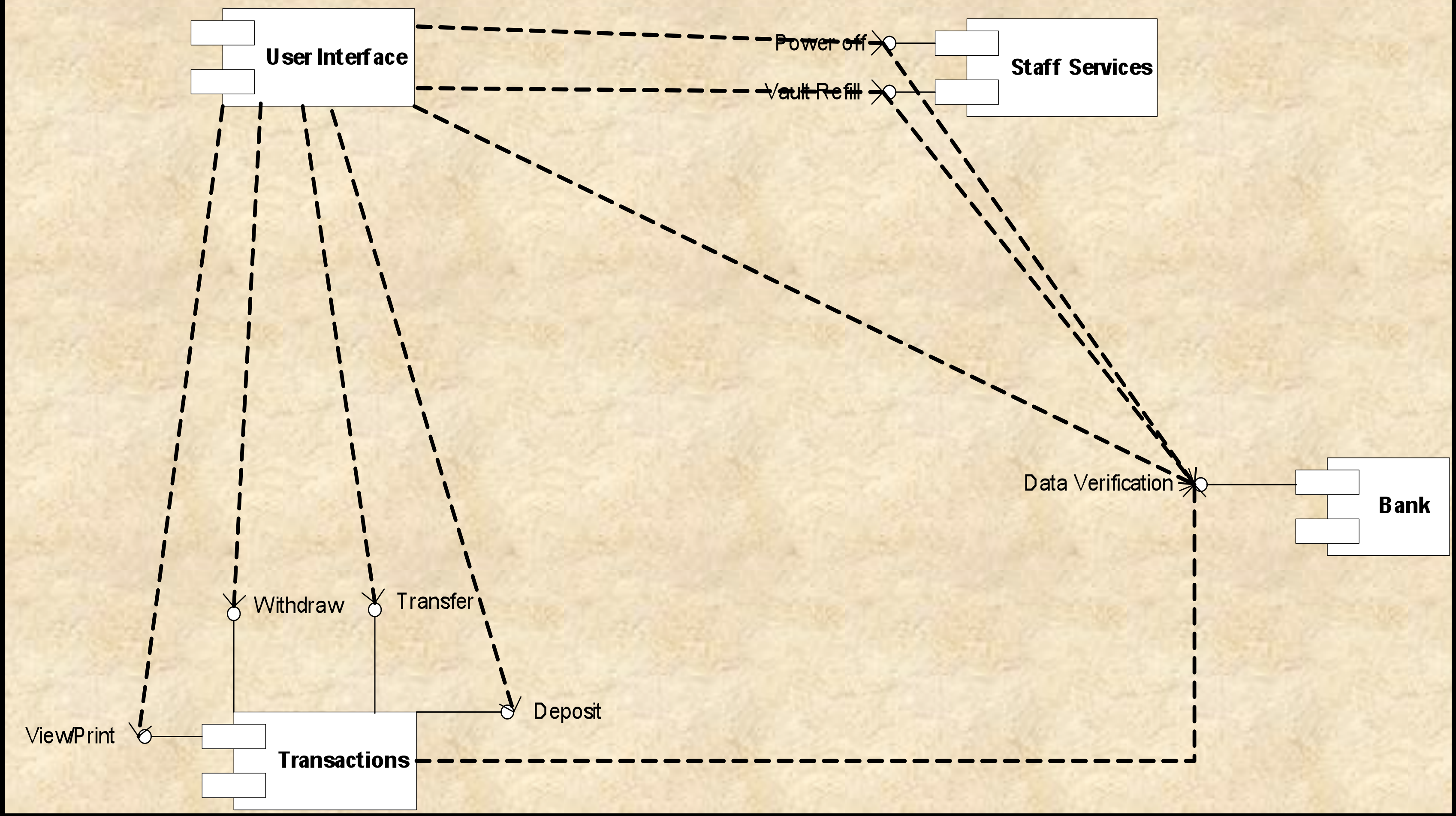
Design and Specification (ABM)

- High Level
  - Component Diagrams
  - State Diagram
- Low Level
  - Detailed Modules: Functions performed by each module.
  - Individual State Diagrams for each module.

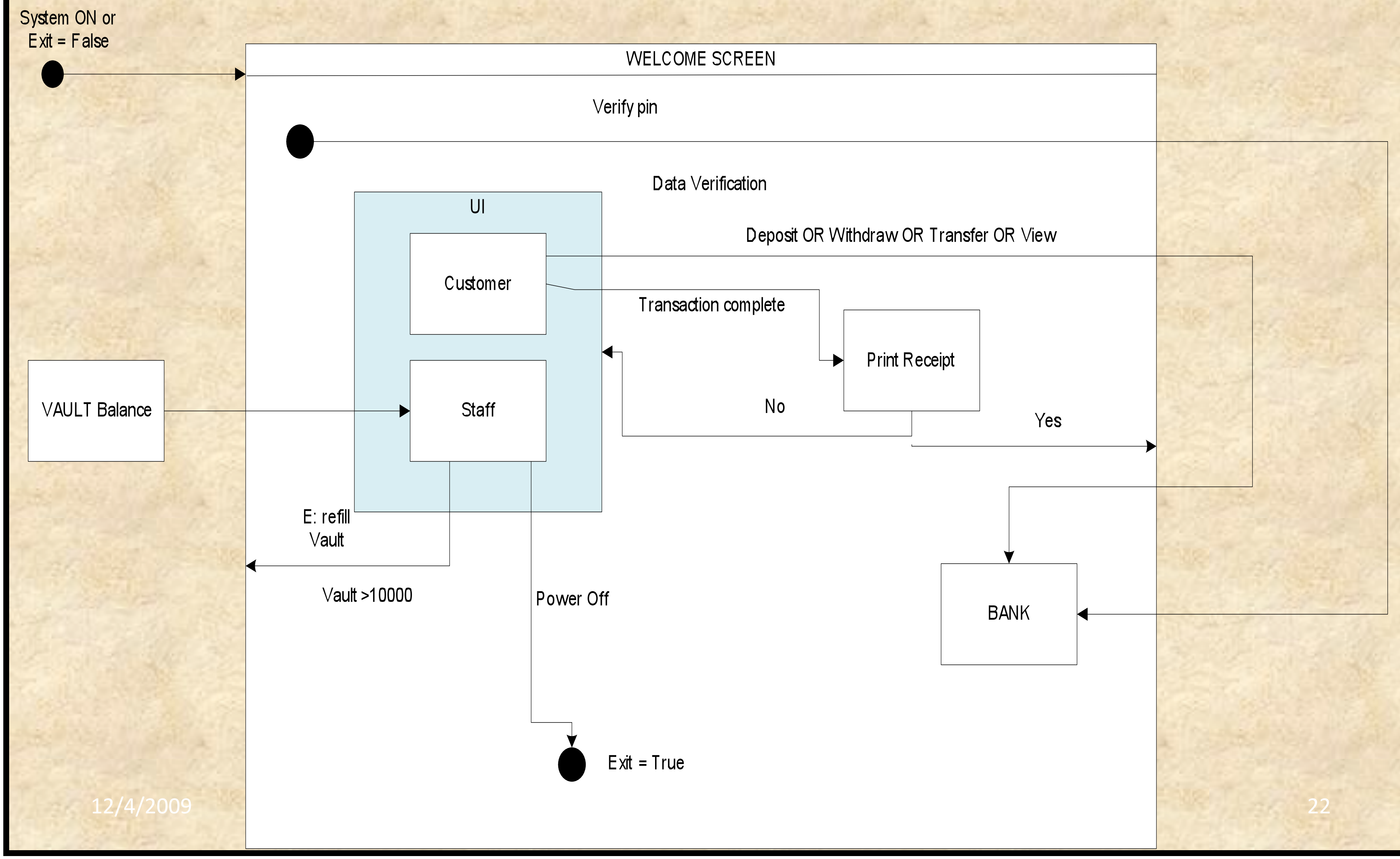
Making of High-Level SDS

- The SRS required the system to interact with two types of users:
  - Staff
  - Customers
- The staff user should be able to perform the following operation:
  - Starting and stopping ABM services.
  - Refill the vault when it is low.
- The customer should be able to perform the following operations:
  - Withdraw money
  - View Balance
  - Transfer Money within accounts.
  - Deposit money
- Also, Bank should perform the following main functions:
  - Verify card and PIN number

Component Diagram for ABM



State Chart Diagram



High-Level to Low-Level SDS

- Listed activities and entities
- Split up similar activities (functionalities)
- High Cohesion, Low Coupling
- Address functions and respective variables
- Classes
- Real life modularity

Code and Module Testing

- Actual code is produced.
- Individual modules are tested
- ABM Implementation:
  - Bank
  - Transactions
  - User Interface

Integration and System Testing

- Individually tested modules put together.
- ABM implementation

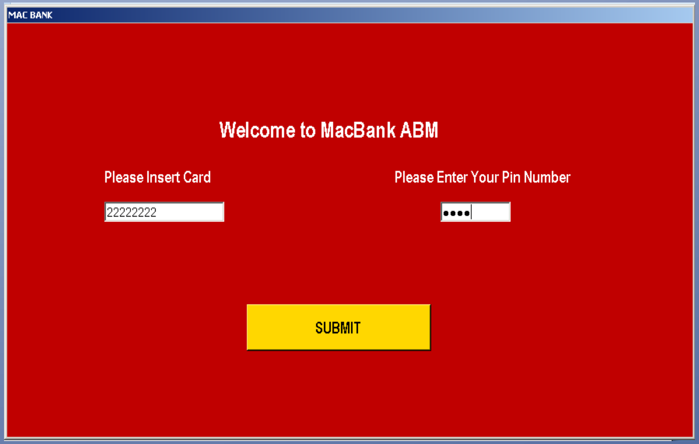
Delivery and Maintenance

- Most important part of designing a software system
- Delivered to the customer after all the tests are passed.
- Modifications come under maintenance.
- ABM Implementation – out of the scope of the course

ABM Implementation

User Interface

- Visual Studio (C#)
- GUI vs. Command Line
- Calling on different classes
  - Forms
  - Limited access to information
- User Functions

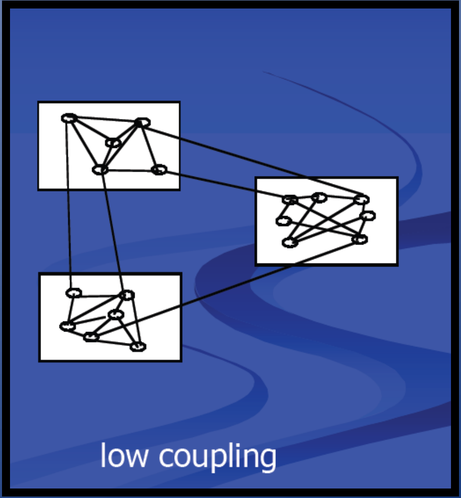


22222222

••••

Transactions Class

- **Basic Functions:** Deposit, Withdraw, Transfer, Balance Inquiry.
- **Supporting functions:** Account Initiliazer, Get Balance, Set Balance.
- **Modularized format.**
- **High cohesion Low coupling between classes.**



Bank Class

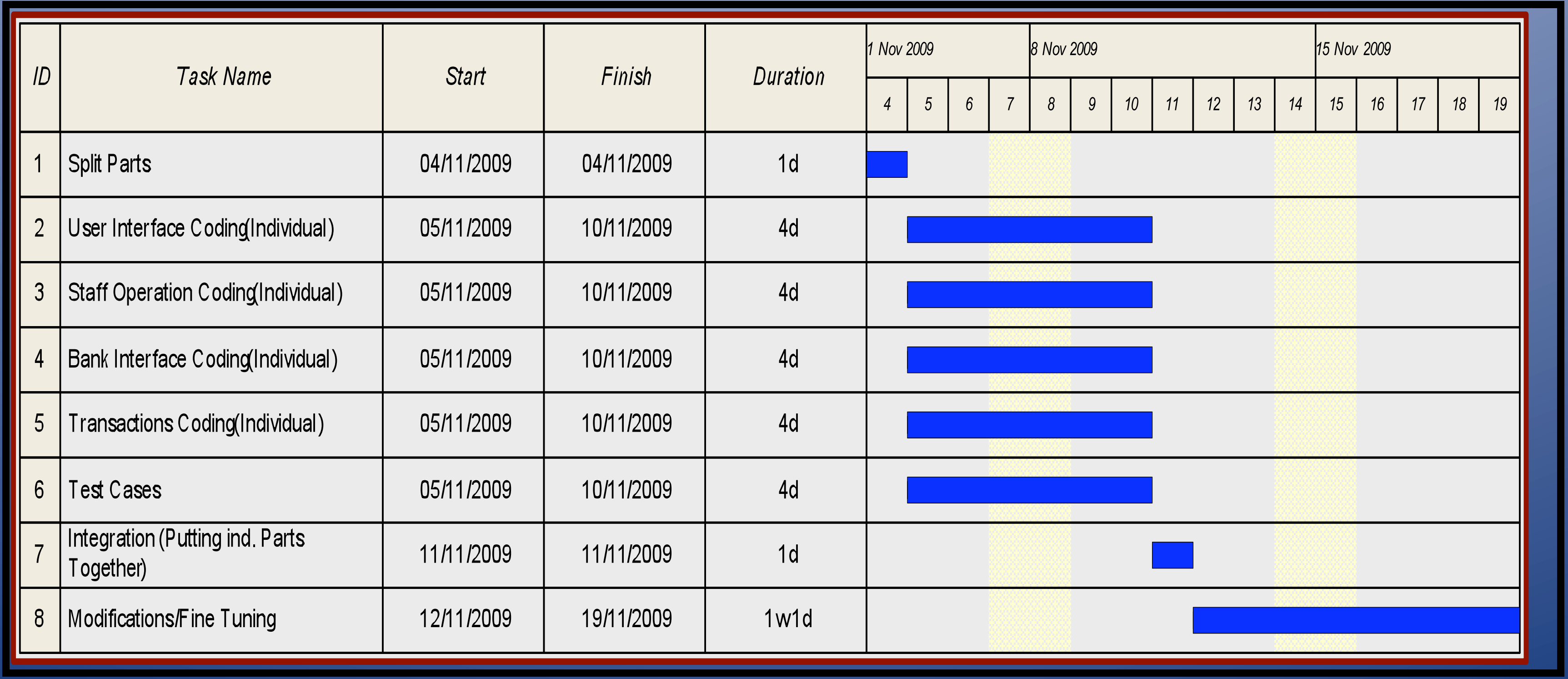
- Functions included:
  - verify_pin
  - verify_card
  - Get_Balance
  - Set_Balance
  - Get_Deposit
  - Set_deposit
  - Vault_amount
- GemBox Application

DEMO

Team Work

- Efficient division of Tasks.
- Weekly meetings to plan next step and follow up on progress
- Dedicated/Passionate team members
- Organizational Tools : GANTT Chart.

Organizational Tools: GANTT Chart



ID	Task Name	Start	Finish	Duration
1	Split Parts	04/11/2009	04/11/2009	1d
2	User Interface Coding(Individual)	05/11/2009	10/11/2009	4d
3	Staff Operation Coding(Individual)	05/11/2009	10/11/2009	4d
4	Bank Interface Coding(Individual)	05/11/2009	10/11/2009	4d
5	Transactions Coding(Individual)	05/11/2009	10/11/2009	4d
6	Test Cases	05/11/2009	10/11/2009	4d
7	Integration (Putting ind. Parts Together)	11/11/2009	11/11/2009	1d
8	Modifications/Fine Tuning	12/11/2009	19/11/2009	1w1d

Gains from the course

- New perception of software design (not just coding).
- Transformed **basic requirements** into a **software product.**
- How to manage projects
- Most practical course ever taken; saw material in action.

Requirements

Time Breakdown

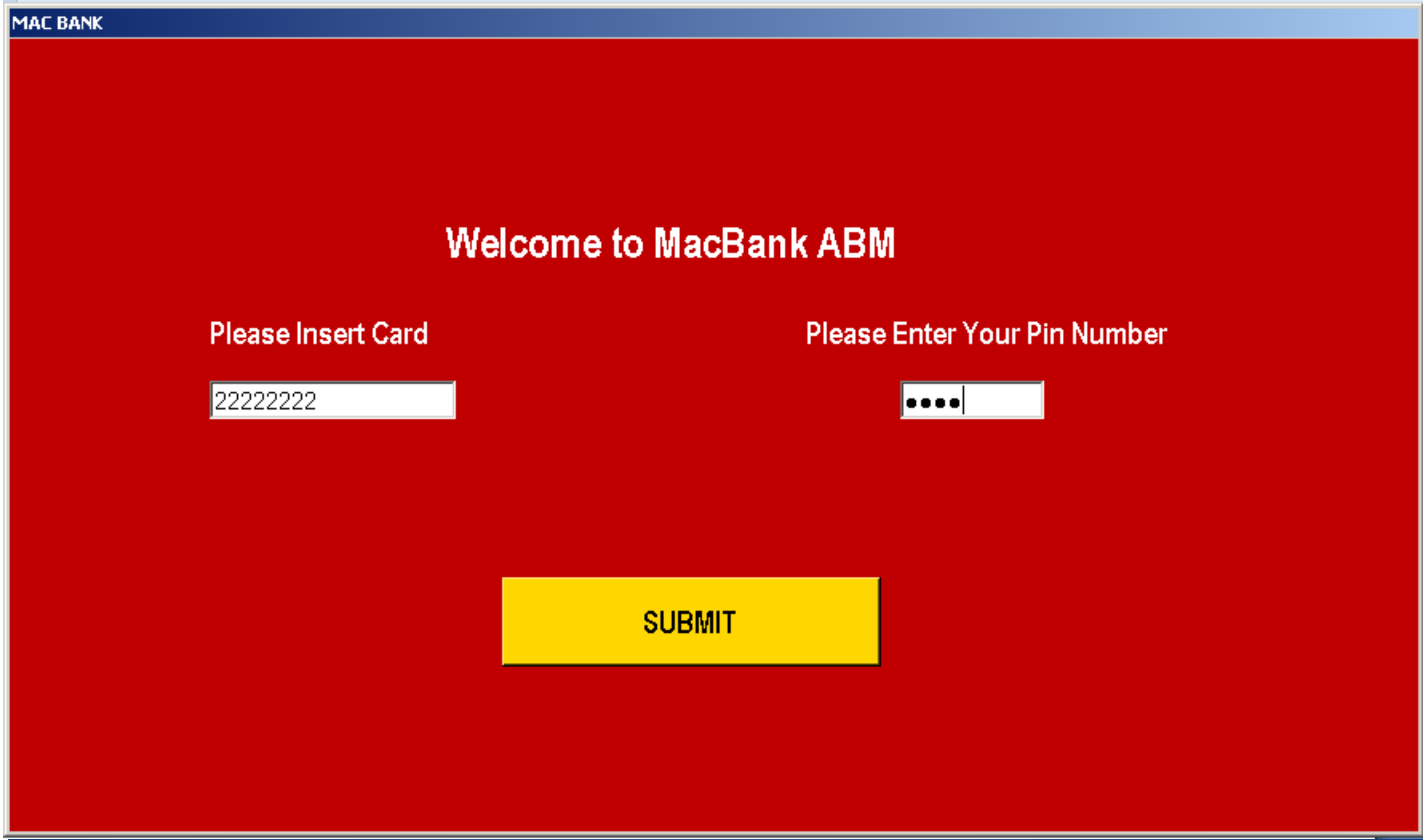
Activity	Time (hours)
SRS	5
SDS High level	10
SDS low Level	10
Coding	30
Debugging	10
Other	5
Total	70

Comments and Suggestions

- More parallelism between lectures and labs.
- More time for implementation.
- **Overall great learning experience!**

Thank You!

Questions?



MAC BANK

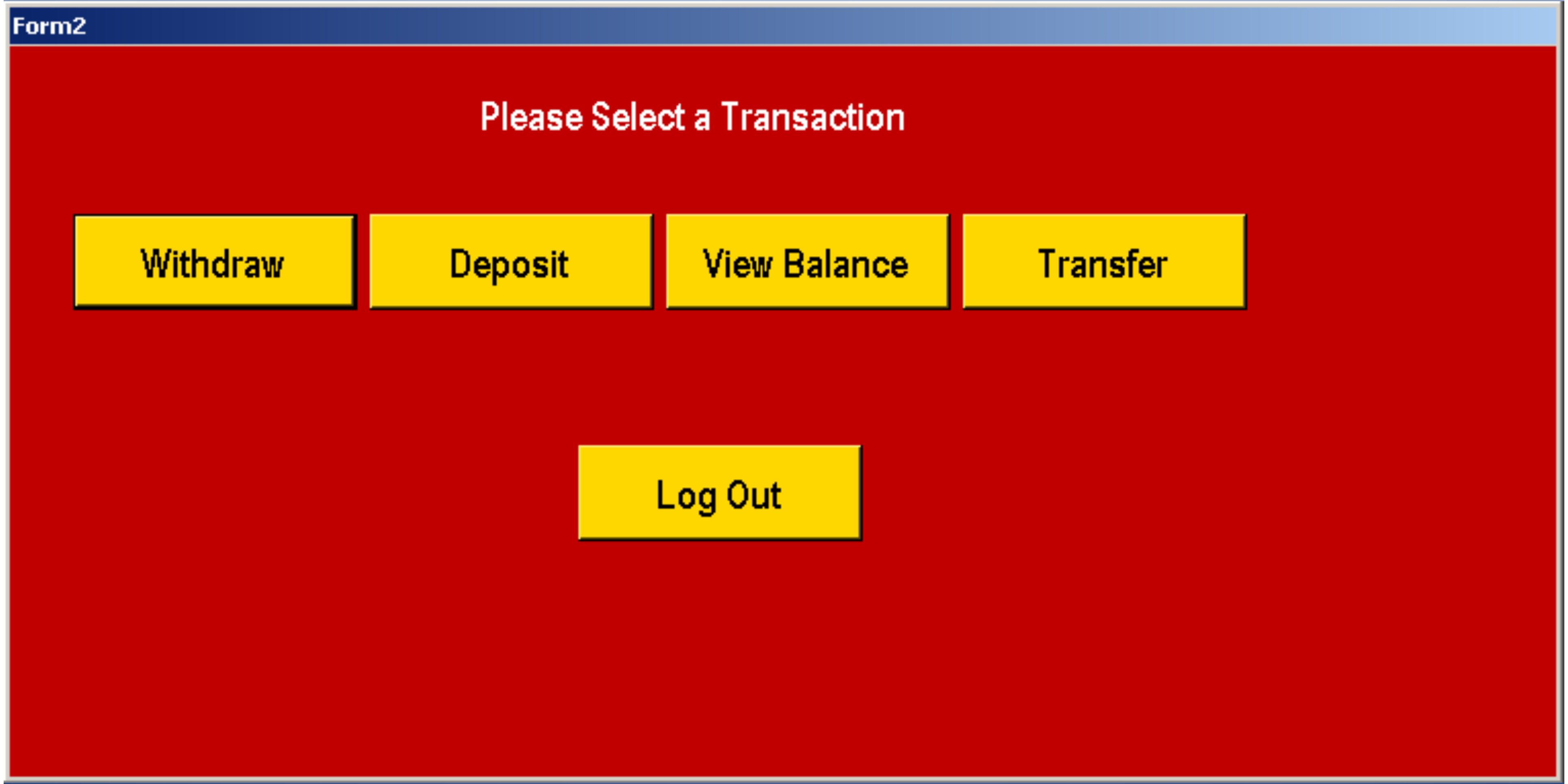
Welcome to MacBank ABM

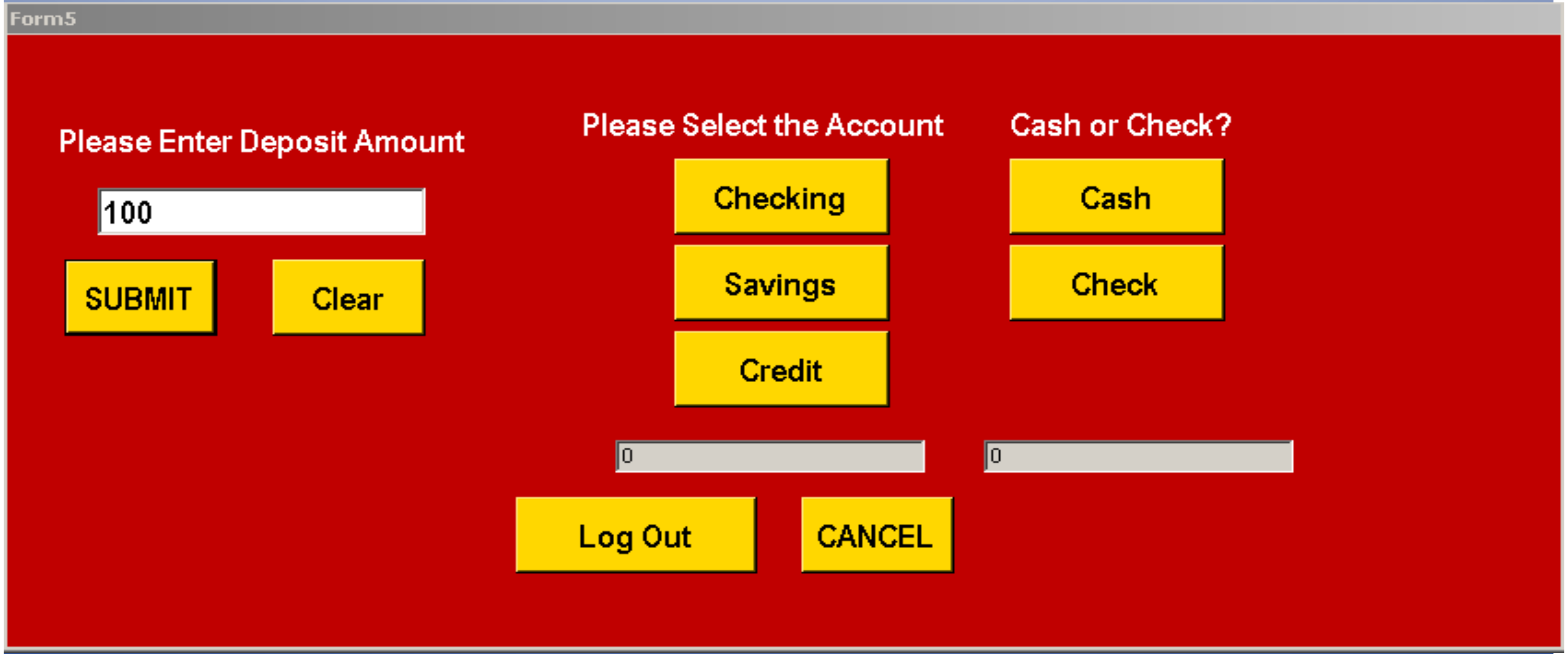
Please Insert Card

22222222

Please Enter Your Pin Number

••••

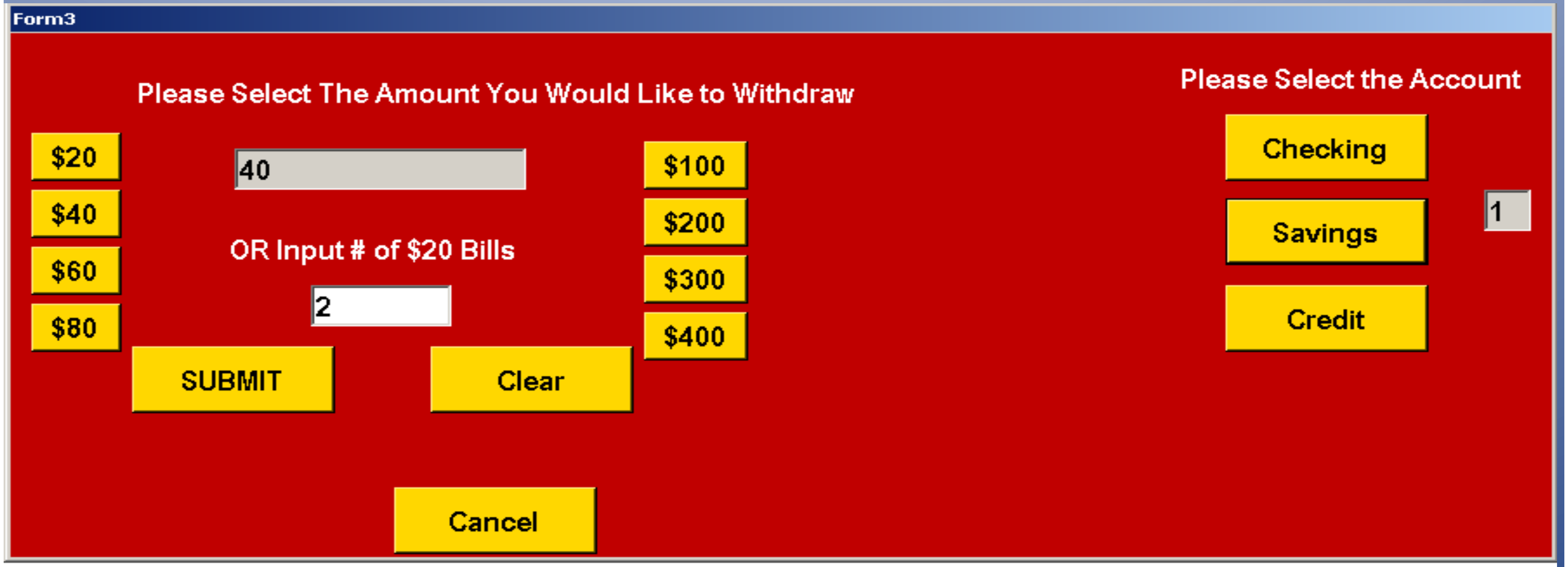

Form2
Please Select a Transaction
Withdraw
Deposit
View Balance
Transfer
Log Out


Form5
Please Enter Deposit Amount
SUBMIT
Clear
Please Select the Account
Checking
Savings
Credit
Cash or Check?
Cash
Check
Log Out
CANCEL

100

0

0


Form3

Please Select The Amount You Would Like to Withdraw

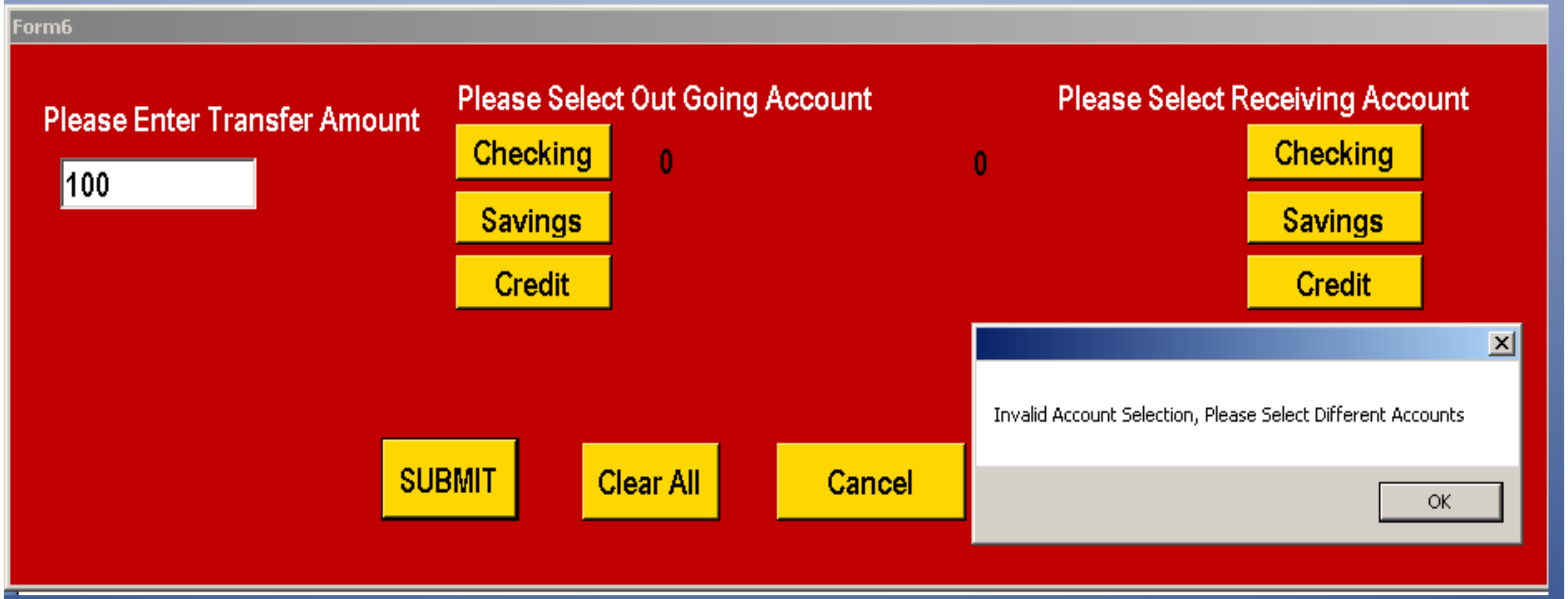
40

OR Input # of $20 Bills

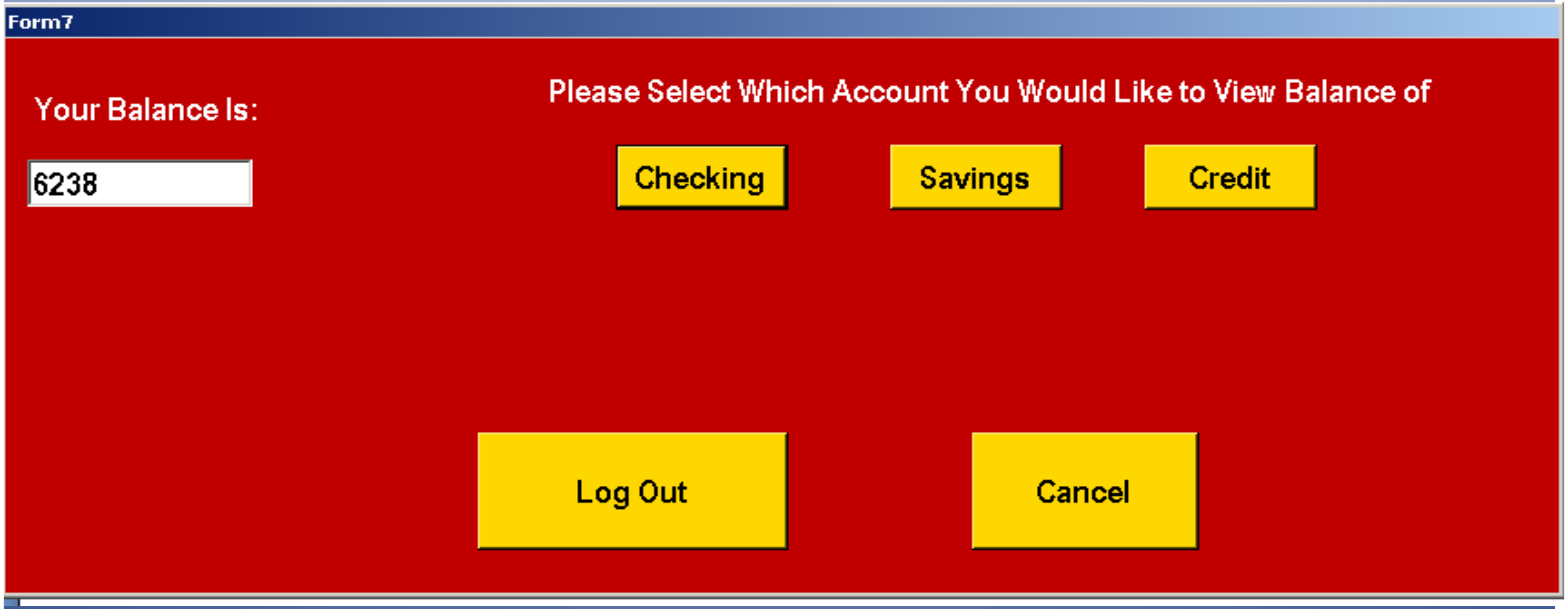
2

Please Select the Account

1


Form6
Please Enter Transfer Amount
100
Please Select Out Going Account
Checking
Savings
Credit
0
0
Please Select Receiving Account
Checking
Savings
Credit
SUBMIT
Clear All
Cancel
OK

Invalid Account Selection, Please Select Different Accounts

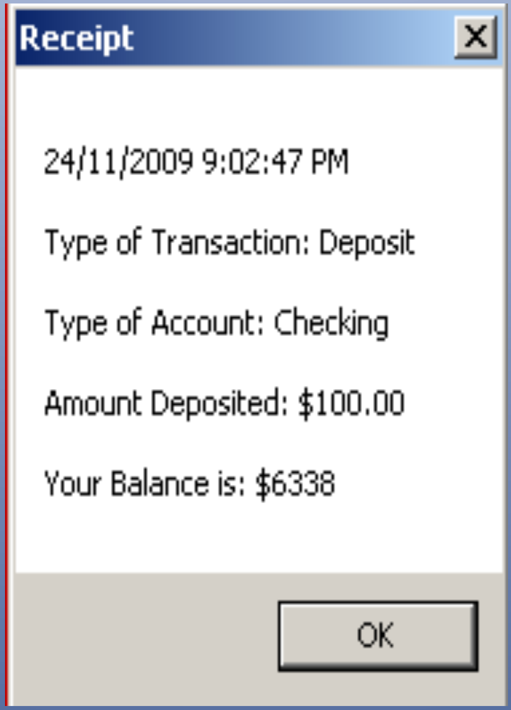


Form7

Your Balance Is:

6238

Please Select Which Account You Would Like to View Balance of



24/11/2009 9:02:47 PM

Type of Transaction: Deposit

Type of Account: Checking

Amount Deposited: $100.00

Your Balance is: $6338