MACBANK ABM

Group 304: Matthew Gardner, Steven Li and Robert Lombardi

Software Eng 3K04
Dr. Kamran Sartipi
December 2, 2009

OVERVIEW OF SOFTWARE ENGINEERING

- "Multi-person construction of multi-version software" [Parnas 1978]
- "Software engineering is an engineering discipline which is concerned with all aspects of software production"
- Solve real problems with computer software, using appropriate tools and techniques as well as specified constraints
- Maintain ethical practices

WHY SOFTWARE ENGINEERING?

- Solve the software crisis
  - Development and maintenance of large and complex software systems
  - For every 8 large projects, 2 are cancelled
  - Majority of projects go over time by 50%
  - 75% of large systems malfunction after implementation
  - 50% of effort is spent on bug testing and fixing
  - System maintenance is 60% of cost
- Provides a disciplined process and methodologies to design, implement, and maintain large systems

SOFTWARE LIFE CYCLE PROCESS

- From inception of an idea for a product to the end of its life

- In General
  - Requirements gathering and analysis
  - Architecture design and specification
  - Coding and testing
  - Delivery and deployment
  - Maintenance and evolution
  - Retirement

LIFE CYCLE PROCESS - WATERFALL MODEL

- Feasibility study
- Requirements
- Design
- Coding and module testing
- Integration and system testing
- Delivery, deployment, and maintenance
- Properties
  - No feedback
  - No parallelism
  - A single delivery date

REQUIREMENT SPECIFICATION

- Benefits:
  - Prevents misunderstandings between producer and user
  - Prevents misunderstanding between engineers and domain experts about affects on the control function to be implemented
- Qualities:
  - Precise, clear, unambiguous
  - Consistent and complete
  - Incremental

MacBank ABM

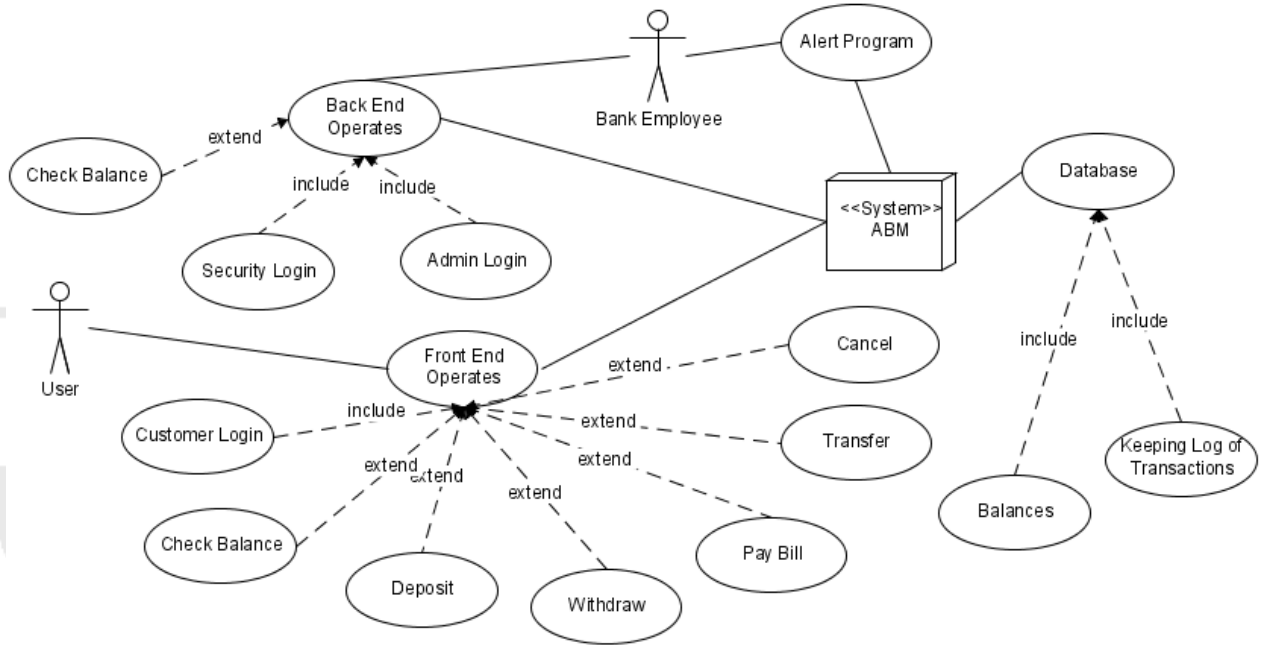
SOFTWARE REQUIREMENTS SPECIFICATION
FROM REQUEST FOR PROPOSAL

- Introduction to Document
  - Formatting Conventions
  - Naming Conventions for Project
- Project Description
  - Perspective
  - Product Functionality
  - Data Flow Diagram
  - Users
  - Constraints
- Specific Requirements
  - User Interface
  - Hardware Interfaces
  - Software/Functional/Behaviour Requirements

MacBank ABM

SOFTWARE REQUIREMENTS SPECIFICATION

- Behaviour Requirements, Use Case View

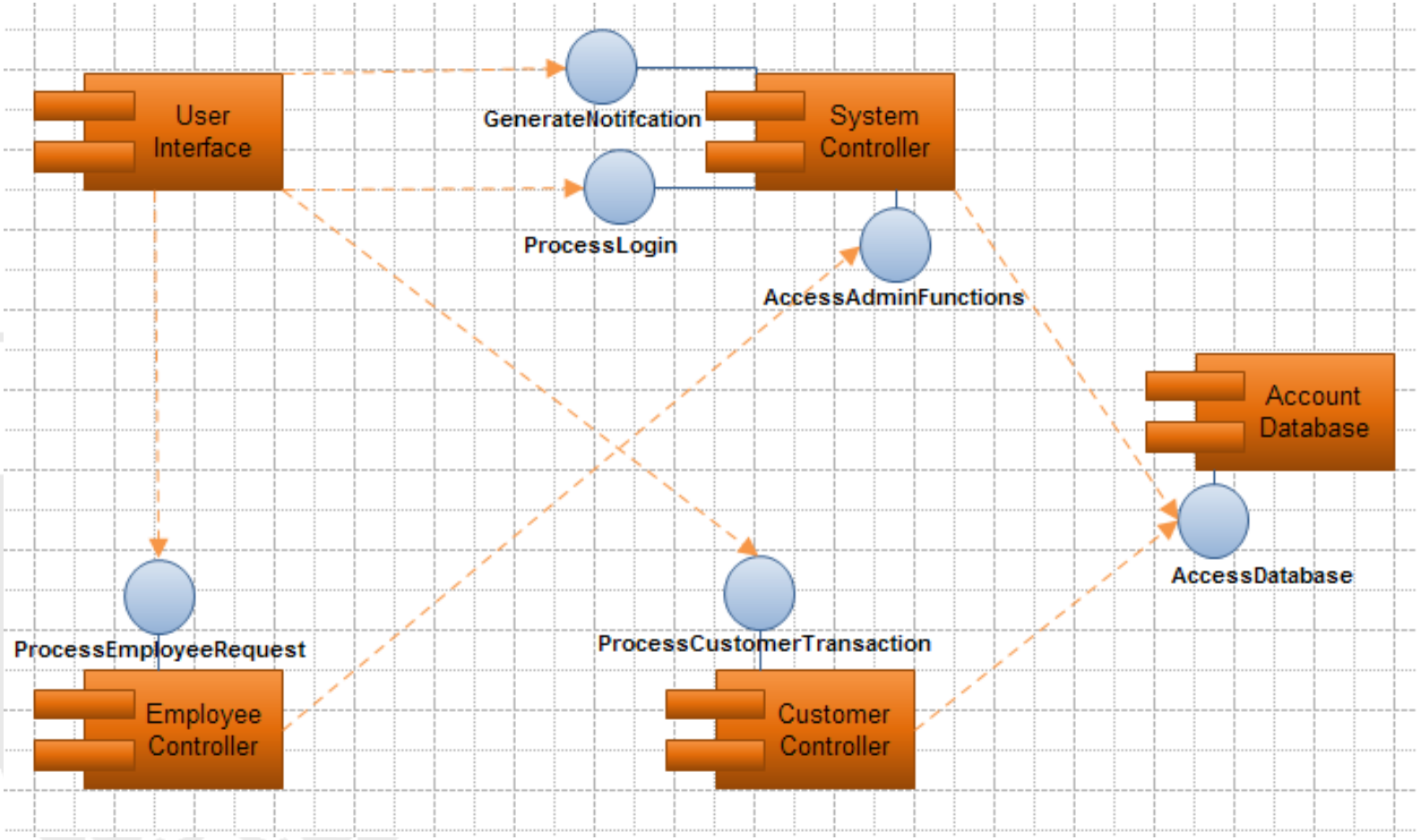


- Non-Functional Requirements
  - Performance
  - Safety and Security
  - Software Quality Attributes

SOFTWARE DESIGN SPECIFICATION

- Broke down system requirements from the SRS
  - Abstract functional modules
- Divided the system into these components:
  - User Interface
  - Employee Controller
  - Customer Controller
  - System Controller
  - Account Database
- Led us to create interfaces to support required interactions

COMPONENT DIAGRAM

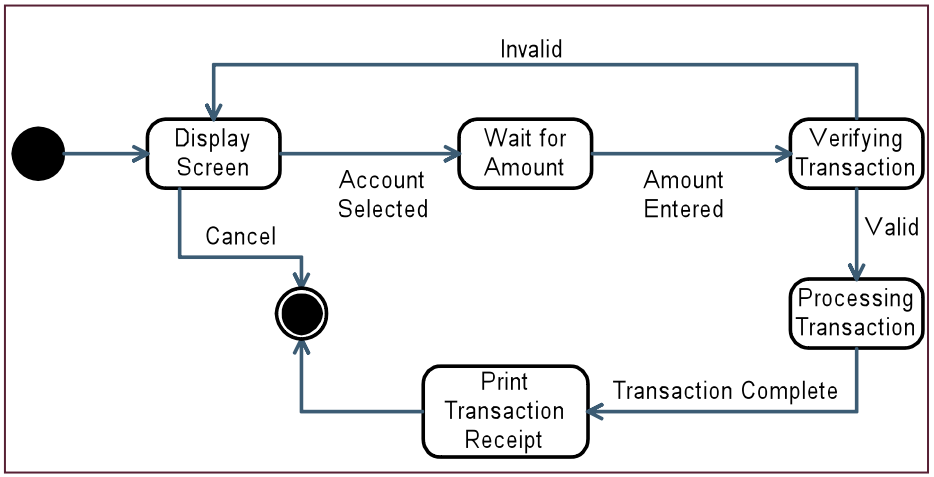


HIGH-LEVEL SOFTWARE DESIGN SPECIFICATION (SDS) FEATURES

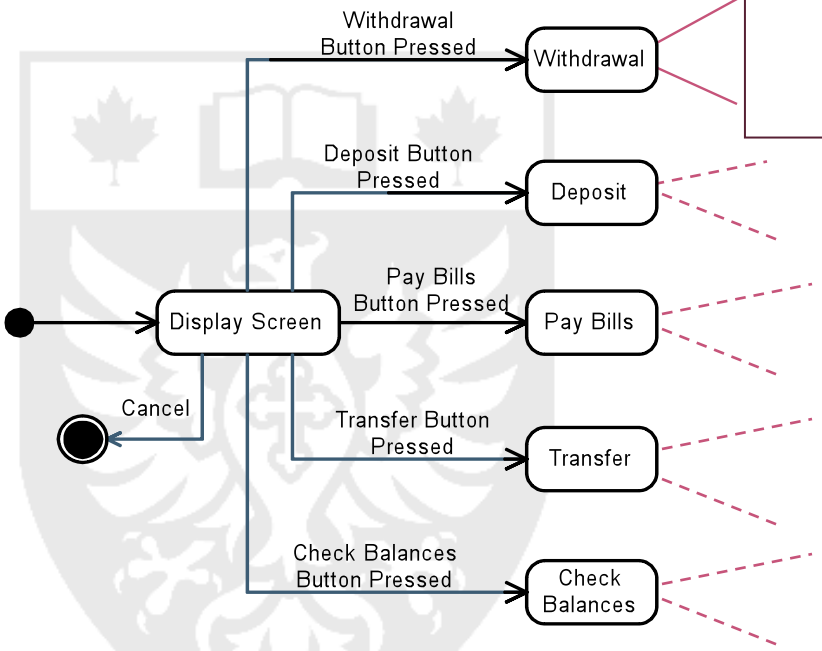
- Outlines features of all components
  - Dependencies described
  - Interfaces described
  - States described
- Statecharts are included for all states of the ABM software

MacBank ABM

HIGH-LEVEL STATECHARTS



Withdraw

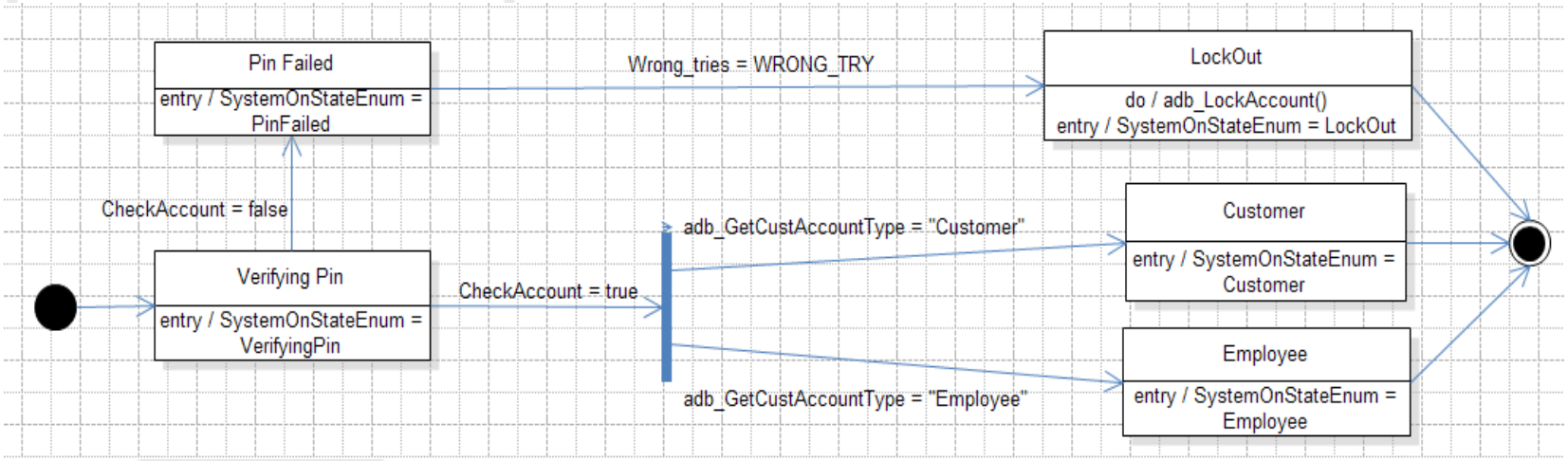


Customer Menu

MacBank ABM

REFINING THE SDS

- Broke down high-level SDS into low-level SDS statechart diagrams
  - More detailed than previous state charts
- Diagrams describe the events that cause the state changes



REFINING THE SDS

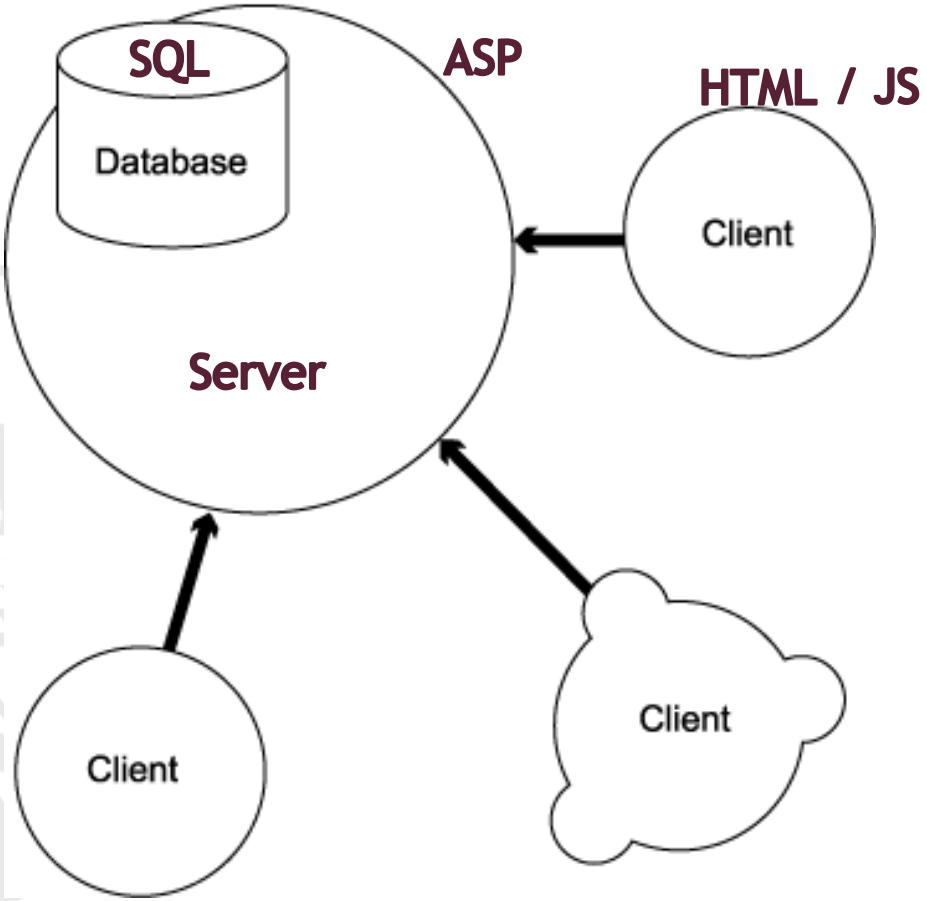
- Describe all Modules in detail
  - Used External Modules and Data Types
  - Internal State Variables
  - Internal Constants
  - Exported Functions
  - Internal Functions
- Focus on how system components would interact and interface

OUR IMPLEMENTATION

- Web-Based
  - HyperText Markup Language (HTML)
  - JavaScript
  - Active Server Pages (ASP)
  - Structured Query Language (SQL)
- SQL database
  - Created using Microsoft Access
- Graphical User Interface (GUI)
  - Adobe Photoshop, Illustrator

HOW IT ALL WORKS

- Web system based on client/server model



THE CLIENT

- The ABM terminal is the client in our implementation (Front End)
- Used for user interface display
- Uses HTML forms to submit data to the server
- Current mode of form submission is “get”
  - Transfers data in plain text using web address
  - Allows easier debugging of system
  - Ideally should use “post” submission method
    - Post sends form items in http message which can be secured using https

THE SERVER

- Internet Information Services (IIS) provides web server back end
- ASP can retrieve HTML form items through the built in request object
  - Get uses request.querystring (“ItemName”) to obtain web address variables
  - Post uses request.form(“ItemName”) to obtain http message variables
- ASP has built in objects and methods to create an object that can interact with a database

THE HTML

- Able to use images generated by Adobe Photoshop for interface
- Adobe slices images up for buttons and textboxes that are html ready
- Each web page acts as a sub-state from the high level SDS design
- Use HTML forms to submit data to ASP for processing
- Add functionality and interaction with JavaScript and ASP

THE JAVASCRIPT

- Provides direct access to HTML form items through the Document Object Model (DOM)
- Allows dynamic processing of form items
- Allows the user to interact with the web page
- Used to create input selection screens for ABM interface
- Used to place input into hidden form variables to pass to ASP

THE ASP

- Allows easy access to/for:
  - HTML form variables
  - MS Access databases
  - Using SQL queries for databases
  - Adding dynamics to user interface
- Makes all decisions for ABM
- Controls are state changes
- Provides the link between HTML/ JavaScript and SQL

THE SQL

- Assuming many records need to be maintained, databases are the most efficient method of storing and retrieving the data
- Database initially created with Microsoft Access
- Use ASP built in Component Object Model (COM) for databases to open a session of communication with database through SQL

OUR IMPLEMENTATION

- Pros
  - Efficient, less code
  - Simple style
  - Easy to understand
  - GUI modifications are simple
  - Accomplished requirements
- Cons
  - Vulnerable to security threats in an unsecure network
  - Real life example would run on a very secure intranet only

DEMONSTRATION

Our ABM System

MacBank ABM

CONCLUSION

- Teamwork
  - Split up work to capitalize on strengths
  - Check each others progress and edit documents and code for others, fresh eyes see mistakes
  - Team meetings should only be held when progress has been made
- Gains and Career Help
  - Knowledge of the design process and document types we would otherwise not know about
  - Experiencing the full process from design to testing is beneficial and unique
  - Our most involved and time-consuming project to date, more realistic in representing work conditions

CONCLUSION

- Gains and Career Help (continued)
  - List as a major project on resumes
  - Can show potential employers that we have knowledge in Software Engineering besides pure coding skills
- Suggestions
  - More time for implementation (coding) and testing
  - Some mock hardware to interface with to add realism

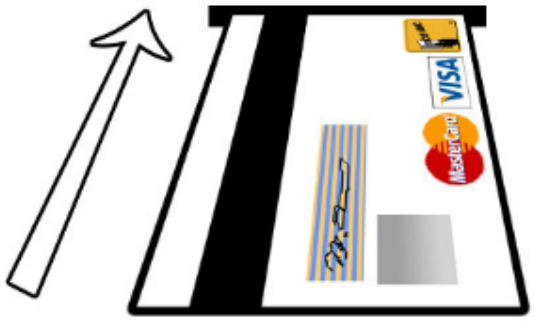
MacBank ABM

QUESTIONS?

MacBank ABM

SOFTWARE DEMO

Welcome to MacBank



Please insert your card

?

Please Enter Your Account Number

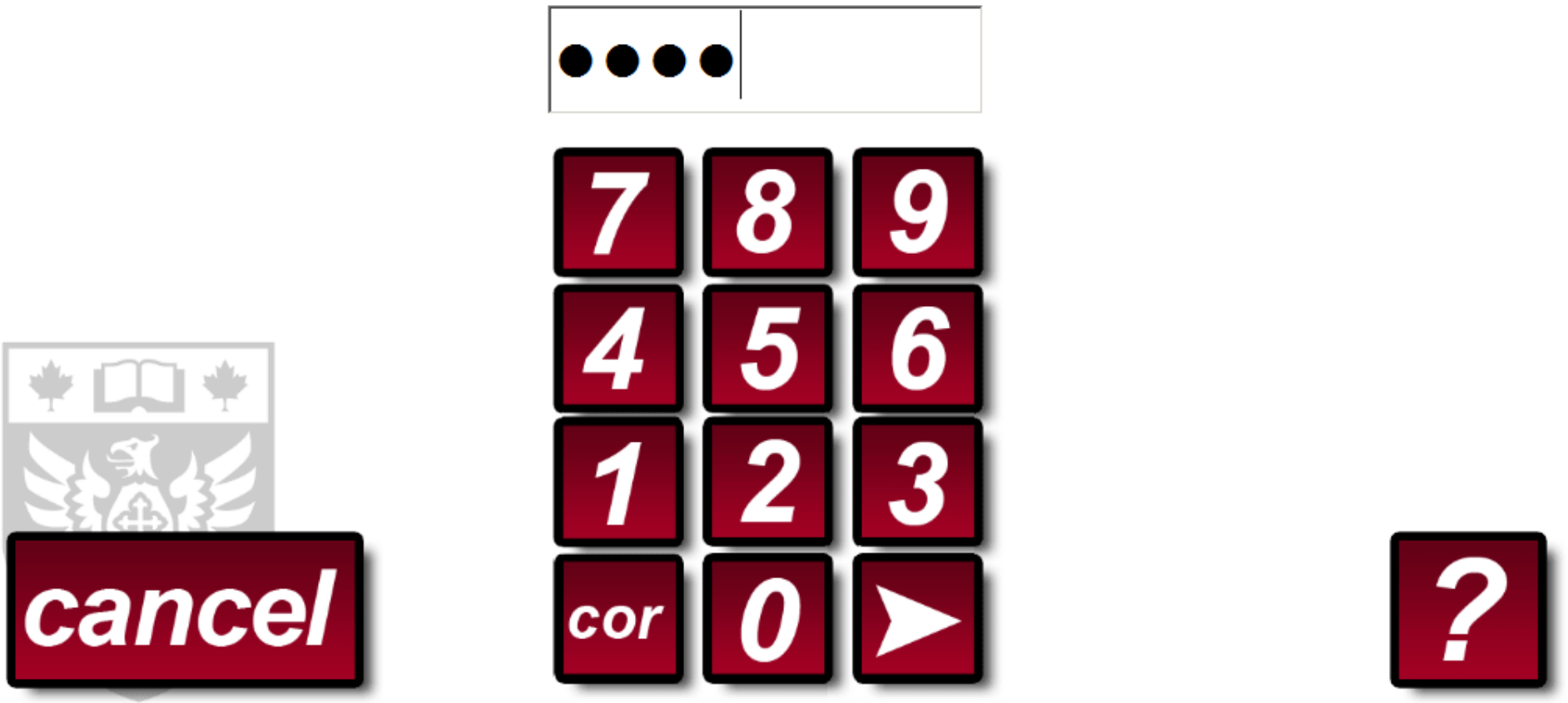
12345678

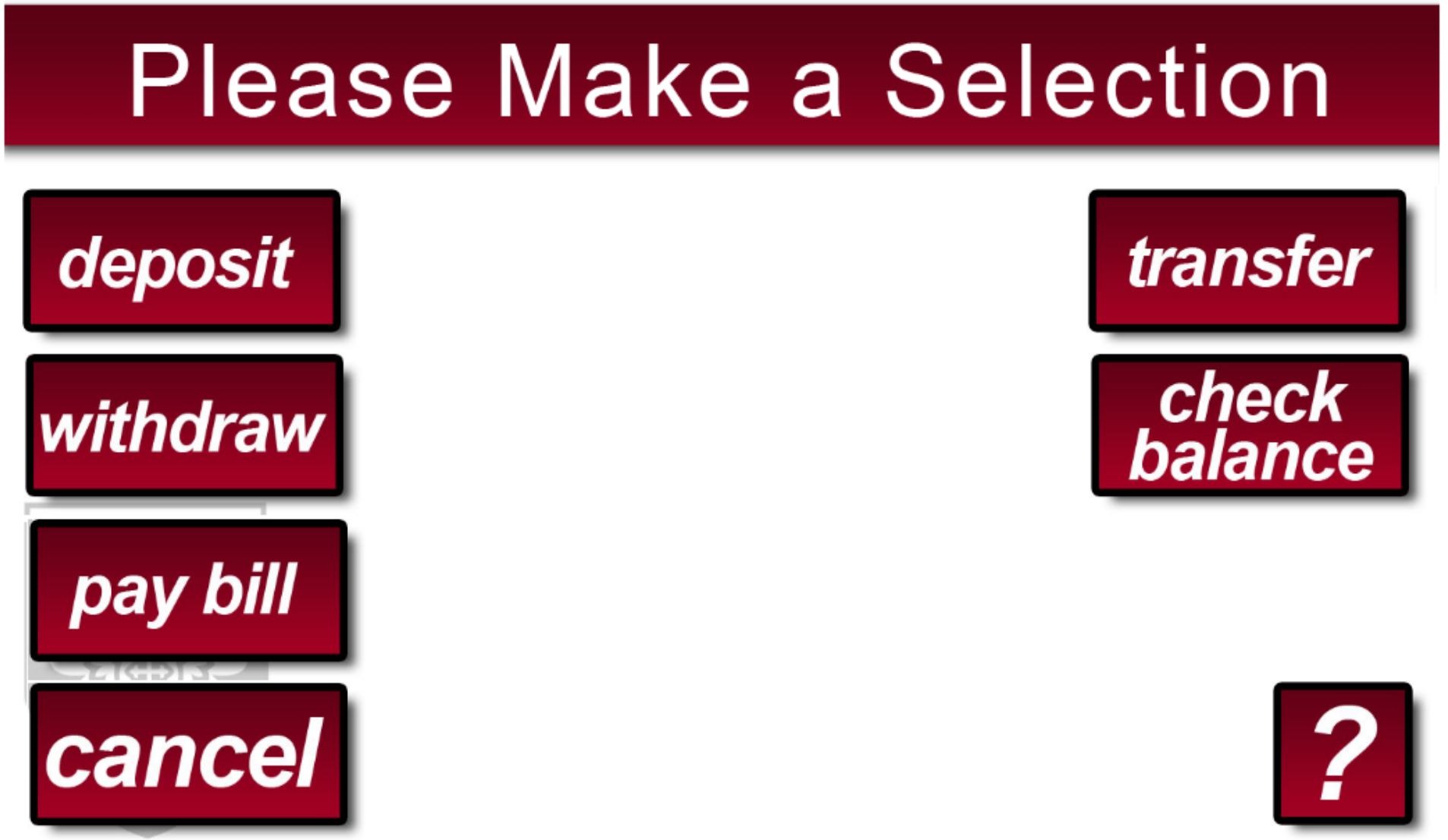
cancel

7	8	9
4	5	6
1	2	3
cor	0	➤

?

Please Enter Your PIN




Please Make a Selection
deposit
transfer
withdraw
check balance
pay bill
cancel
?

Please Enter Deposit Amount

500.00

cancel

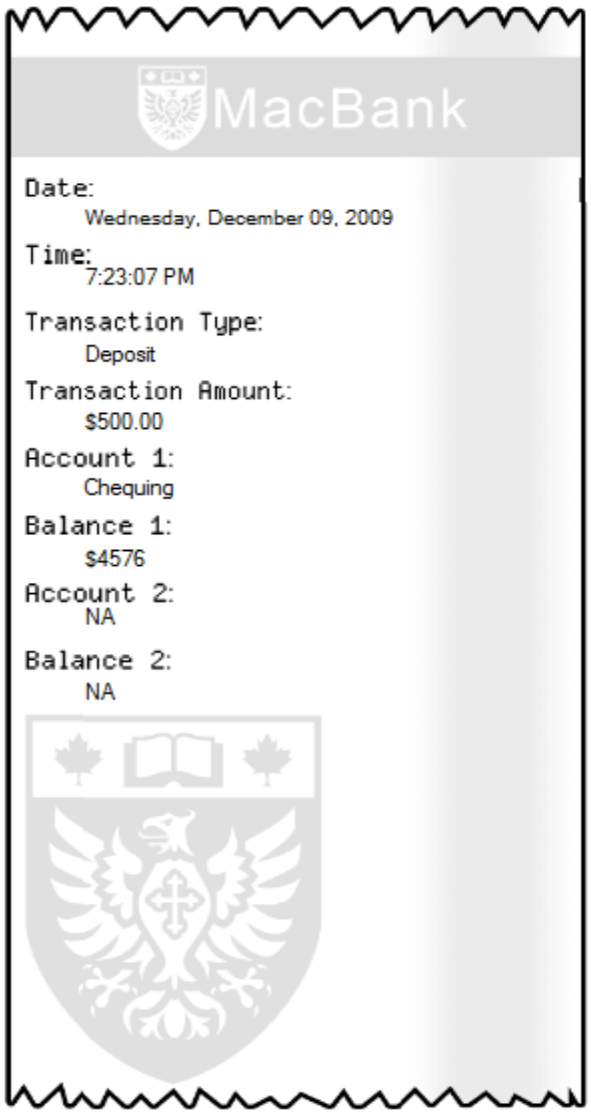
7 8 9

4 5 6

1 2 3

cor 0 ➤

?


MacBank
Date:
Wednesday, December 09, 2009
Time:
7:23:07 PM
Transaction Type:
Deposit
Transaction Amount:
$500.00
Account 1:
Chequing
Balance 1:
$4576
Account 2:
NA
Balance 2:
NA

Dispose

Employee Selections

ABM Status

The current ABM Balance is $33900

OK

?