Project Analysis

CMSC 495 6380

6/5/2021

Group 5

Summer Smith, Sean Dwyer, Keith Combs

**Input Data**

* Login credentials
  + User ID
  + Password
* Registration information
  + User ID
  + Password
  + Email
* Money transactions
  + Choice of transaction (transfer, deposit, or withdraw)
  + Deposit amount
  + Withdraw amount
  + Transfer amount
  + Transaction confirmation or cancellation

**Output Data**

* Login status
* Transaction status (pending, canceled, complete)
* Account status
  + Savings balance
  + Checking balance
  + Interest percentages
  + Interest earned
* Alerts
  + Negative withdraw or transfer
  + Request transaction confirmation

**Data Processing**

* Register new user ID, password hash, and email with new account (values set to 0)
  + Write new account record to database if successful
  + Display registration failure if user ID or email collides with existing accounts
  + Display registration failure if password fails to meet complexity criteria
  + Display registration success if registered
* Authenticate logon user ID and password hash against registered accounts
  + Display logon failure if user ID or password hash does not match
  + Display logon success if matched
* Track user authentication by webpage session
* Execute transaction based on user input:
  + Display request for confirmation or cancellation of transaction
  + Trigger execution based on user confirmation
  + Increase or decrease account by user input value for deposit or withdrawal respectively
  + Increase one account and decrease other account by user input value for transfer transaction
  + Retrieve balance of accounts and interest percentage and earnings for account status
  + Display resulting balances or requested information to user with transaction confirmation
  + Display alert for requested value resulting in overdrawn account, prompt for confirmation or cancellation
  + Continue with transaction if confirmed

**Context Diagram**

**Subsystem Diagram**

**Subsystem Descriptions**

* Authentication Subsystem
  + Registration Subsystem
    - Accepts user input for user ID, password, and email
    - Compares user ID and email against existing account records
    - Displays registration failure if user ID or email matches existing account
    - Analyzes password for appropriate complexity
    - Displays registration failure if password is insufficiently complex
    - Writes new account record to database with user provided user ID, password, and email, and with initial savings and checking values set to zero
    - Displays registration success
  + Logon Subsystem
    - Accepts user input for user ID and password
    - Compares user ID against existing account records
    - Displays logon failure if user ID does not match any existing account
    - Compares password hash to saved password hash of matched existing account
    - Displays logon failure if password hash does not match
    - Displays logon success
  + Session Subsystem
    - Tracks logon session
    - Provides logon verification to other subsystems before proceeding with transactions
* Account Subsystem
  + ???

**Subsystem Mapping**

**Enhancements**

* Alerts for large withdraw or transfer (amount set by user)
* Transfer money between different users

**Risk and Risk Mitigation**

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Name** | **Description** |
| 6/5 | Summer | Created document with cover page, revision history table, and outline. Completed Input Data and Output Data sections. Added one possible enhancement. |
| 6/5 | Sean | Added Data Processing section and initial Subsystem descriptions. Added one additional enhancement. |