Online ATM System: Part II

CMPE131: Introduction to Software Engineering

Spring 2020

Team 1

Siya Bedi
Carmen Bradfield
Jay Brahmbhatt
Duong Cao
Marco Caruso
Casey Delaney
Anmol Dhoor

Low-Level Design

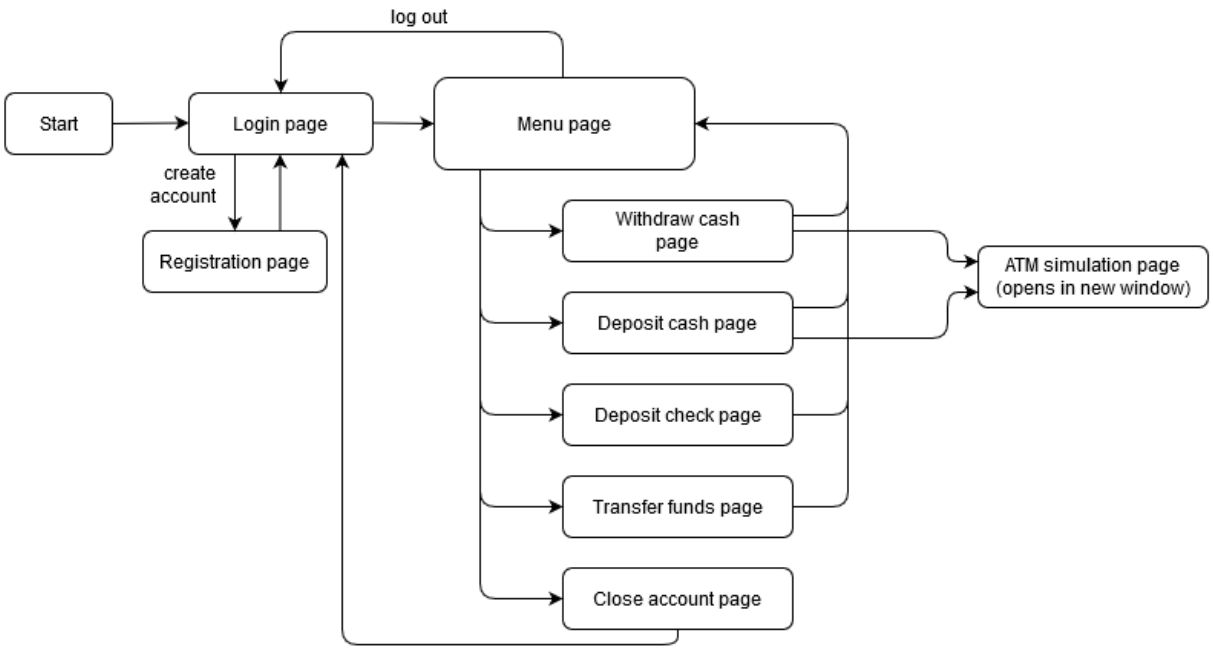

***Figure 1**:* Page relationship diagram

Page-by-page breakdown

1. *Login (open to this page)*
   - Login form
     - Username
     - Password
   - "Forgot your password" option -> redirect to page x (optional functionality)
     - Addition of security questions to registration and database storage
     - Ask for name, e-mail, security questions
   - Main scenarios
     - Correct Password and username
       - -> redirect to page 3
     - Incorrect Password and username
       - -> Display error message
       - Possibly notify user if more than 3 incorrect attempts were made
         - Retrieve email from DB corresponding to username and send an email
   - "Create account" option -> redirect to page 2

2. *Registration page*
   - Registration form
     - Username
     - Name

        - Email
            - If an account with this email already exists -> Display Error Message (highlight text in red)
        - Password
        - Re-enter password
            - If passwords do not match -> Display Error Message (highlight text in red)
            - If password is not valid -> Display Error Message (highlight text in red)
- On successful registration -> redirect to page 1
    - Inputs for registration

3. *Menu page*
- "Withdraw Cash" option -> redirect to page 4
- "Deposit Cash" option -> redirect to page 5
- "Deposit Check" option -> redirect to page 6
- "Transfer Funds" option -> redirect to page 7
- "Log Out" option -> performs log-out, redirect to page 1
- "Close Account" option -> redirect to page 8

4. *Withdraw cash page*
- Withdrawal process
    - ATM Location Function
    - Enter the amount of money you intend to withdraw
        - This amount should be checked against balance & other withdrawal requests (call available balance function)
            - If balance is not available, call the balance not available function to display an error message
        - Calls Subtract Money Function if balance available function returns true
    - Tell user that this information is being shared to the selected ATM
    - Generate a random code for the user to authorize their withdrawal at the ATM
    - Open ATM simulation in a new window
- List of pending withdrawal request(s)
    - Ability to cancel withdrawal

5. *Deposit cash page*
- Deposit process
    - ATM Location Function
    - Enter the amount of money you intend to deposit
        - Calls add money function
    - Tell user that this information is being shared to the selected ATM
    - Generate a random code for the user to authorize their deposit at the ATM
- List of pending deposit orders(s)
    - Ability to cancel deposit

6. *Deposit check page*

- Image submission box
- Check deposit form
    - Some of this information should be stored so that the same check cannot be used again

7. *Transfer funds page*
- Transfer form
    - Typical testing scenarios: correct information for the receiving account, sufficient funds for transfer, etc
- Calls available balance function
    - If balance is a available, proceed with transaction
    - Else, display a warning to the user by calling balance to available function

8. *Close account page*
- Panel that shows whether or not the account can be closed yet, cannot close if:
    - Current balance is not 0
        - Direct user to Deposit (page 5) or Transfer Funds (page 7)
    - Pending withdrawals/deposits
        - Direct user to Withdraw (page 4) or Deposit (page 5)
- Close account form
    - Require that user re-verify their information

ATM Simulation
- Login form
- Withdrawal/Deposit authentication form
- Show verification

Other tasks
1. Check Database
    a. Database for images of check
    b. Link each image to a specific user
    c. Save routing number on check
2. Add check Function
    a. Calls retrieve user information function
    b. Adds image of check corresponding to the user to the check database
3. Retrieve user information Function
    a. When a user's information is needed, call this function
    b. Example: Adding money to user account, retrieve user information and then perform a task
4. Logout Button
    a. When a user is logged in, this button will appear on any page to allow the user to logout
5. Select an ATM location Function
    a. Function that can be implemented onto pages to choose an atm location
    b. List of ATM locations

        i. User selects one
    c. Output is a single ATM
    d. For use with deposit and withdrawal pages
6. Available balance function
    a. Calls the user information function
    b. Takes an input of money to withdraw
    c. Checks input against balance and pending withdrawal requests
    d. Boolean output
        i. True = Money is available
        ii. False = Not enough money: calls balance not available function
7. Balance not available function
    a. If balance available function returns false, this function will be called
        i. Displays a warning message to the user
8. Menu header
    a. Menu tabs that contain links to pages
9. Confirmation prompt
    a. When leaving a page with a form that has unsaved data, ask the user if they are sure they want to leave
10. Styles
    a. Create a universal style for the website to follow
    b. Example: Rounded corners on buttons
    c. Color scheme (limit to about 3/4, including background, primary, accent)
    d. Font
11. Add Money function
    a. Takes an amount as an input
    b. Void output
    c. Updates the user profile account by the corresponding amount
12. Subtract Money Function
    a. Takes an amount as an input
    b. Retrieves money from account
    c. Money output
        i. Update the user database and return the amount withdrawn
    d. This function can only be called if the available balance function returns as true
13. Forms
    a. Create form pages to be used on the registration page and login page
        i. Securely Link forms to MongoDB through a pipeline
14. ATM Location Database
    a. Stores locations of ATM's
    b. Returns a single ATM location when the user has selected one

High-level Test Plan

Success Criteria:

Task	Home Page / Login Page
Goal	Give basic information; Allow user to log into their account
Summary	1. Describe name, logo, services offered, etc. 2. Ensure login form is filled out correctly 3. Allow user to create account if needed

Task	Registration Page
Goal	To create a new bank account
Summary	1. Ensure form is filled out correctly 2. Store data for a new account in the database

Task	Menu Page
Goal	To provide a central location
Summary	1. Create an attractive central hub linking the user to the other pages

Task	Withdraw Cash Page
Goal	Allow user to authorize withdrawals and view pending withdrawals
Summary	1. Ensure user has required balance to make the withdrawal they want 2. Give user a random authorization code for their withdrawal 3. View and cancel pending withdrawal requests

Task	Deposit Cash Page
Goal	Allow user to authorize deposits and view pending deposits
Summary	1. Give user a random authorization code for their deposit 2. View and cancel pending deposit requests

Task	Deposit Check Page
Goal	Allow user to deposit checks
Summary	1. Able to upload image(s) of check 2. Ensure form is filled out correctly (& not repeated)

Task	Transfer Funds Page
Goal	Allow user to transfer funds to a different account
Summary	1. Ensure form is filled out correctly

Task	Close Account Page
Goal	Allow user to shut down their account
Summary	1. Check if this account can be closed 2. Ensure user verifies their information correctly

Testing Descriptions & Scenarios

- ATM Simulation
  - User consistently inserts the incorrect PIN
  - Cannot find the authorization code given by user
- Login page
  - User tries to login with the wrong password too many times
- Create Account page
  - Username is already in use
  - Password does not match the Re-enter Password field
  - Password is not strong enough
  - Email is already in use
- Withdraw Cash page
  - User attempts to withdraw more money than they have (check against balance & pending withdrawals, but not against pending deposits)
  - User attempts to withdraw more money than is allowed at one time (>$2000 pending, regardless of balance)
- Deposit Check page
  - User submits a check that has already been used
- Transfer Funds page
  - User tries to transfer more funds than they have (check against balance & pending withdrawals, but not against pending deposits)
  - User tries to transfer funds to an account that does not exist
- Close Account page

    - User fills out their verification information wrong
- Functional Testing
    - Individually testing each unit of code
    - Integrating with rest of the codebase
- Performance and Load Testing
    - Making sure that the software runtime on the machine is not slow
    - Response time from API calls should be minimum
    - Compatibility to scale the platform if number of user increases
- Security Testing
    - Protecting customer's information

Tools

- MEAN (MongoDB, ExpressJS, AngularJS, NodeJS)
- GitHub (Software Team Collaboration)
- WhatsApp (Team Communication)
- Google Drive (Documentation)

Methodology

- Agile
    - Agile methodology is a great tool that helps monitor and control the progress of the project. By breaking down the big task into smaller tasks before working on it, it will help increase the productivity and the effectiveness of the team overall.
        - The team breaks down the high-level design into details which can be called as low-level design. The software team involves software and computer engineering students who are responsible for describing the functions, features, and test strategies.
        - The team discusses the project every 2 or 3 days in order to keep track of the progress as well as solve any roadblocks which may happen.