Design Document for Coin Control

Group IP-103

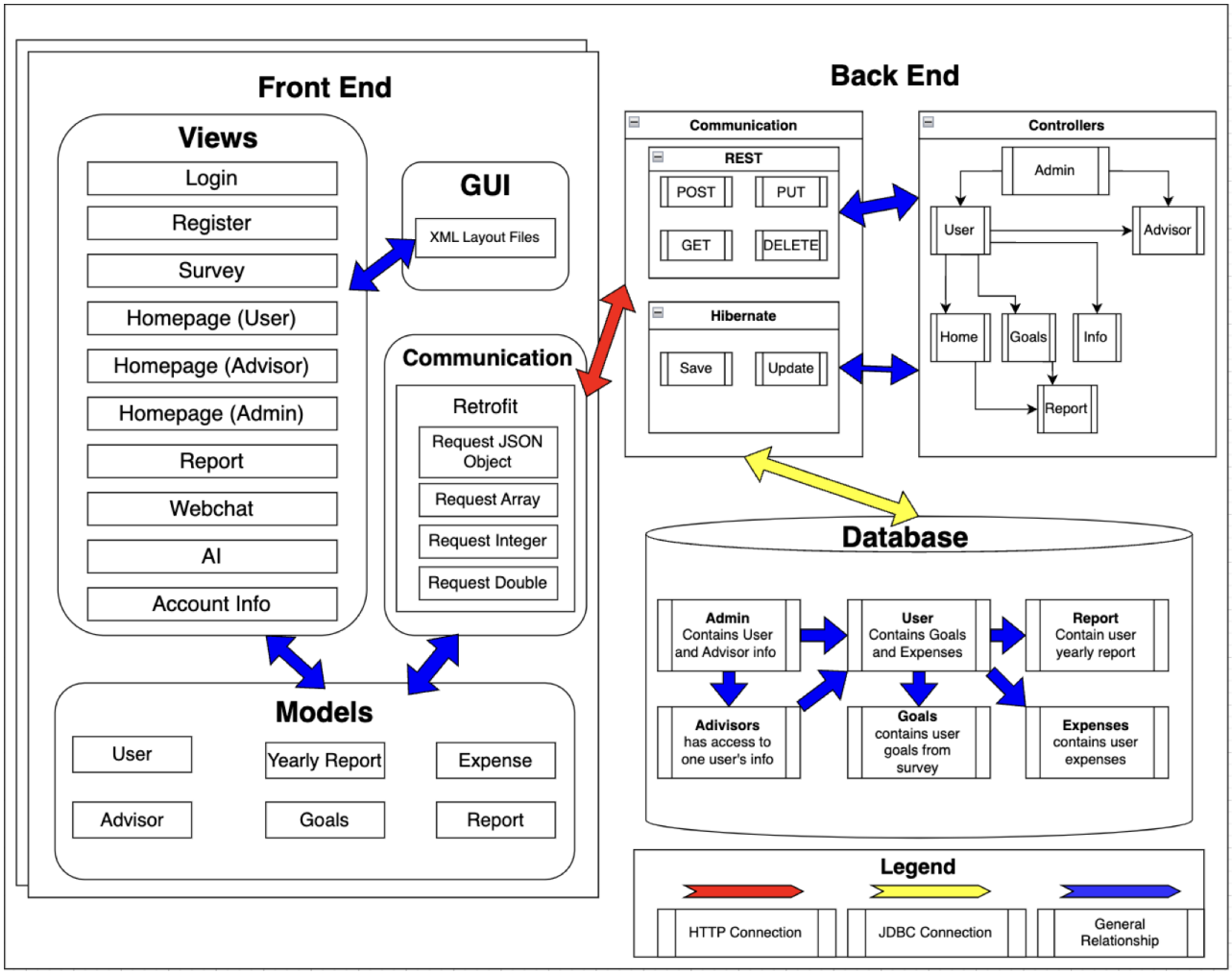
Kareem Eljaam: 25% contribution

Andrew Gooding: 25% contribution

Osaid Samman: 25% contribution

William Griner: 25% contribution

Block Diagram



Frontend

Account Info

- Displays the users Account Info and allows them to edit that info.
- The current username, password, and email, are obtained via GET call and displayed in EditText boxes.
- Username, password, and email all have a button next to them. If the user presses this button, whatever is in the EditText box will be sent to the database via PUT call.

Admin Page

- At the top of the page, there are two buttons, ‘show users’ and ‘show advisors’. Each button will show a list of users or advisors upon being clicked.
- At the bottom of the page, there is an EditText box where the username of a user who is going to be deleted can be typed. Below that is a button that upon being clicked, will delete the user in the database who’s username was typed into the EditText box, via DELETE call.

Advisor Home

- This page has the coin control logo at the top of it, and two progress bars. These two bars utilize GET calls to display how much the client has spent on their weekly allowance, and how close they are in reaching their weekly savings goal.

Chat

- There is an EditText box for the user to type his message. Upon clicking the ‘send’ button, websockets are utilized to send, receive, and display the message for the users involved. The messages are displayed in scrollView.

Create Account

- This activity uses multiple EditText boxes to gather information from the user such as: first name, last name, email, username, password, and whether or not they are an advisor. Upon the ‘Create’ button being clicked, a series of checks are conducted: whether the username inputted is already being used, and if the email inputted is already in use. If both of those are false, a POST call is made to create a user in the database.

Home

- This page starts with the Coin Control logo at the top. Next, the user can input an expense into an EditText box. Upon clicking a button to submit this expense, a PUT call is made. This call updates the allowance and savings bars, subtracting from the expense them both.

Login

- This page has two EditText boxes, one for username and one for password, as well as a 'login' button. Upon clicking 'log in' two POST calls are made, one for a regular user login and one for an advisor login. A series of checks are conducted in both: Are the username and password correct, and is the account recognized at all in the database? Upon a successful login, the user is taken to the homepage

SurveyInfo

- The User sets a time estimate of 6 months or 1 year by clicking a button for either one, inputs their monthly income, current bank account balance, and desired bank account balance. Upon clicking the 'submit' button, a POST call is made, that sends this information to the database.

YearlyReport

- Initial Balance, Desired Balance, Total Expenses, and Total savings are all displayed on this page using GET calls.

Backend

Communication

The backend uses mappings to update the database based on information sent to the given mappings' URLs. These include:

• Post: send information on an item to be added to the database.
• Get: request information, often with an identifier for the specific item requested from the database.
• Put: send information to update a specific item in the database.
• Delete: send an identifier to delete a specific item from the database.

Controllers

The controllers contain the mappings for communication between frontend and the database. These include:

- User: Manages user-related operations, including registration and login.Mappings:
  - /user/all: Retrieves a list of all users.
  - /finduser/{id}: Retrieves a user by ID.
  - /findusername/{username}: Retrieves a user by username.
  - /user/register: Registers a new user and associates them with an admin.
  - /user/login: Authenticates a user.
- Advisor: Manages advisor-related operations, including registration and login.
  - /advisor/all: Retrieves a list of all advisors.
  - /findAdvisor/{id}: Retrieves an advisor by ID.
  - /findAdvisorName/{username}: Retrieves an advisor by username.
  - /advisor/register: Registers a new advisor.
  - /advisor/login: Authenticates an advisor.
  - /advisor/user/{username}: Retrieves the total expenses of an advisee.

- Admin: Manages admin-related operations.
    - /admin/allUsers: Retrieves usernames of all users associated with the admin.
    - /admin/all/Advisor: Would retrieve usernames of all advisors associated with the admin.
    - /admin/delete/user/{userName}: Deletes a user by username.
- Goal: Manages user goals.
    - /goals/all: Retrieves all user goals.
    - /goals/{id}: Retrieves a goal by ID.
    - /goals/post: Posts new user goals.
- Homepage: Manages user expenses and provides information for the homepage.
    - /weeklyExpenseAmount/{weekNumber}: Retrieves the total expense amount for a specific week.
    - /currentWeeklyExpenseAmount: Retrieves the total expense amount for the current week.
    - /newExpense: Adds a new expense.
    - /addExpense/{userName}: Adds an expense and updates total weekly expenses and savings.
    - /allExpenses: Retrieves all expenses.
    - /calculateGoals/{userName}: Calculates and returns user goals based on income and expenses.
- AccountInfo: Manages user account information.
    - /currentUser/infoFromID/{id}: Retrieves user information by ID.
    - /currentUser/infoFrom/{username}: Retrieves user information by username.
    - /new/{newUsername}/username/{oldUsername}: Changes the username of a user.
    - /new/password/{password}/{username}: Changes the password of a user.
    - /new/email/{email}/{username}: Changes the email of a user.
    - /user/logout/{username}: Logs out a user by updating their response ID.
- YearlyReport: Manages yearly reports for users, including generating and retrieving reports.
    - /report/{username}: Retrieves or generates a yearly report for a user.

Table Relationships Diagram