# PROBLEM SOLVED

*Describe the problem you have identified and want to solve with your project. Also provide a brief overview of your project and how it will solve the problem.*

Problem: people who are meticulous about tracking their spending habits must add extra steps into their financial tracking workflow to account for shared expenses with friends and family. In addition, without a standardized method of tracking shared expenses and the amount owed by each person in the expense-sharing group, it is easy to lose track of who owes whom, the amount owed, and whether shared expenses have been fully settled. In short, tracking shared expenses is error-prone and tedious.

My project aims to solve these problems by integrating a standardized method for tracking shared expenses into a personal finance app by adding financial workflows for expense-sharing “teams” (friend groups, families, couples, etc.).

Although individual app users may choose to use the app for their own personal finances and will receive all the expected functionality of a standard spending tracker – i.e. a list of all expenses, the ability to add/edit/delete expenses, the ability to categorize expenses into publicly available or personalized categories, and basic analytics about spending habits – users who choose to create or join expense-sharing teams with other app users will unlock the primary functionalities of this app. The main expense sharing features include the ability of any user to:

* Select existing app users to create new teams, and define the percentage that each team member is responsible for paying on all team expenses
* Once on a team, any team member can edit the team (add/remove members, update expense sharing percentages)
* Assign a new expense to a team if the user is currently a member of that team
* View all expenses – personal and shared – in a centralized All Expenses view, with the ability to sort and filter for date, description, and team.
* For any shared expense, a user will see how much they owe to the original payee. Or, if the user is the original payee, they will see how much each team member owes to them. These amounts are automatically calculated by the functionality of the app, based on the percentage assigned to each user in the team.
* For any shared expense with an unpaid balance, a user can pay the original payee to settle the expense.
* For each team, a user will see the net amount (i.e. the collective “debt” amounts for all existing shared expenses on the team) owed to or from each other member on the team.

Stretch goal:

* For any team member to whom the user owes a net balance, a user can pay the team member the net amount to settle many expenses at once.

# WIREFRAMES

*Wireframes must include each view you intend to have. Each view must include a layout to show where content will be displayed.*

<https://miro.com/app/board/uXjVN17SZjY=/?share_link_id=739304517319> (note, there are hyperlinks on each button that automatically snap to the next intended view)

# ERD

*User-related data. At least one many-to-many relationship.*

<https://dbdiagram.io/d/capstone-65aaf18aac844320ae511980>

**A few notes based on Josh’s feedback on the ERD:**

1. Each user gets a personal teamId:

Upon registration, each new user will automatically be assigned to a personal team of 1 with a percentage due for each expense of 100%. By default, this personal team id will be assigned to all expenses entered by the user. The user’s userId will also be automatically assigned to each expense.

If the user assigns the expense to a team, then the expense.teamId will be changed to the associated team id.

By assigning a user to a personal team and using this teamId on all personal expenses, I can use the same workflow for processing and calculating expenses regardless of whether the expense is personal or shared.

1. Keep track of the facts in the database, and do additional calculations at the level of the UI.

Each payment made between users is posted to the payments table, and payments do not affect the data in the expenses tables. Any time the amount owed is shown to the user (either the amount owed for a single expense, or a net amount owed to a team member), this calculation is done based on the data about the expense and any payments that have already been posted.

In other words, amount owed is never hard coded into the database – amount owed is calculated based on the data about the expense and any payments that have been made.

**How I will perform amount owed calculations:**

1. Amount owed by an individual to the original payee for a single expense:

General calculation: Amount Owed = (User’s Fractional Share) - (User’s Amount Paid)

Within an Expense Details view, the app has ready access to: currentUser.id, expense.id, expense.teamId

To calculate a user’s fractional share:

* Determine fractional share the user is responsible for within this team:
  + User = (fetch apiURL/users?id={currentUser.id}&\_*embed=userPayments&\_*embed=userTeams)
  + Current userTeam = user.userTeams.find((userteam) => (userteam.teamId === expense.teamId))
  + Current user’s expense fraction = currentUserTeam.splitFraction
  + Therefore, (User’s Fractional Share) = expense.amount \* currentUserExpenseFraction

To calculate the amount, if any, that a user has already paid towards this expense:

* Determine amount already paid by current user towards this expense
  + Expense = (fetch apiURL/expenses/{expenseId}?\_embed=payments)
  + List of paymentIds associated with current expense = expense.payments.map(p => paymentIds.push(p.id))
  + Already made userPayments = user.userPayments.filter((UP => paymentIds.includes(UP.paymentId))
  + List of paymentIds made by the current user = currentUserPayments.map(p => userPaymetnIds.push(p.id))
  + List of payment amounts made by current user = expense.payments.filter((p => userPaymentIds.includes(p.id))
  + Total amount paid by current user towards this expense = paymentsMade.map(p => amountPaid += p.amount)

THEREFORE

* With user’s fractional share and user’s amount already paid, I can calculate amount owed by subtracting the two

1. Net amount owed to an individual on a team

General calculation for amount owed by the current user to team member A:

((My total fractional share to A) - (Sum I’ve paid to A)) – ((A’s fractional share to me) - (Sum A has paid to me))

If above value is positive, I owe A money. If above value is negative, A owes me money.

This general calculation would be repeated for each team member in order to display current user’s relationship to everyone in the team.

Within a Team Details view, the app has ready access to: currentUser.id, team.id

All userTeams associated with this team = (fetch apiURL/userTeams?\_teamId={team.id}&\_expand=user

All expenses and payments associated with this team =

(fetch apiURL/expenses?teamId={team.id}&\_embed=payments)

To calculate (My fractional share to A):

* Filter all expenses from above for those made by A:
  + expenses.filter((e) => (e.userId === A-user.id))
  + Sum paid towards expenses by A = filteredExpenses.map((e) => amount += e.amount)
* My split fraction for this team =
  + userTeam.splitFraction where userTeam.userId is my currentUser.id
* Therefore, my total fractional share due to A = (sum paid by A) \* (my split fraction)

To calculate (Sum I’ve paid to A):

* Get all payments made by me towards expenses originally paid by A:
  + myPayments = (fetch apiURL/userPayments?userId={currentUser.id}&payeeId={A-user.id}&\_expand=payment)
* Remove payments made by me towards A that were not associated with the current team (i.e. if A and I happen to both be members of a separate team)
  + Expenses made by A: expenses.filter((e) => (e.userId === A-user.id))
  + expenseIds made by A: filteredExpenses.map(e => expenseIds.push(e.id))
  + Filtered userPayments = myUserPayments.filter((up) => (expenseIds.includes(up.payment.expenseId))
* Calculate sum of my user payments to A:
  + filteredUserPayments.map(up => amountPaid += up.payment.amount)

THEREFORE

* With (My fractional share to A) and (Sum I’ve paid to A), I can calculate my net amount owed to A

The last step is to perform the same steps as above from the perspective of A.

* Calculate (A’s fractional share to me) by
  + Filtering all expenses for those made by me
  + Using A’s split fraction for this team to calculate A’s total fractional share due to me for the expenses I paid
* Calculate (Sum A has paid to me) by
  + Getting all payments made by A towards expenses originally paid by me
  + Removing payments that were not associated with the current team
  + Calculating the sum of A payments to me

THEREFORE

* (A’s fractional share to me) - (Sum A has paid to me) = A’s net amount owed to me

FINALLY

* (Net amount owed to A by me) - (Net amount owed to me by A) = Net Debt
  + If positive, I owe A. If negative, A owes me.

# PROJECT MVP: STORIES & CRITERIA

## As a registered user, I should be able to access all app pages after logging in.

**Given** the user wants to see a home page with at-a-glance personal finance summary information,  
**When** the user logs in,  
**Then** they will be directed to the home page, which shows a personal finance snapshot that lists the user’s top 3 spending categories, a shared monthly expenses summary that lists the shared expenses from the last 30 days for teams they are in, and a quick actions section containing links to useful pages in the app (add new expense, view all expenses, create new team, view your teams, customize categories, and view your profile).

## As a user, I should be able to enter and manage a new expense

**Given** the user wants to log a new expense,  
**When** the user clicks Add New Expense on the home page or All Expenses view,   
**Then** they will be redirected to the Add New Expense view. In this view, a blank form is displayed with text input fields for the date, description, and amount of the expense, a category dropdown for selecting the category, and a yes/no dropdown for determining if the expense is shared. If yes (shared) is selected, another dropdown is displayed which contains all teams that the logged in user is associated with. Given the expense may not fit into the currently available categories, the user has the option to create a new category within the Add New Expense form. At the bottom of the form is a Create Expense button, which when clicked, posts the new expense to the api and redirects the user to All Expenses view.

## As a user, I should be able to view all of my expenses

**Given** the user wants to see a list of all personal and shared expenses in a single view,  
**When** the user clicks Your Expenses in the nav bar or View All Expenses in the Quick Actions section of the home page,   
**Then** they will be redirected to the All Expenses view. On the left 2/3rds of the page, a list of all expenses, each of which includes the date, expense amount, description, category, and either shared or personal, is displayed. On the right 1/3rd of the page, a box with the heading, Expense Details, and text, “Select an expense to see more details…” is displayed.

## As a user, I should be able to filter all of my expenses

**Given** the user wants to filter expenses by either description or team,   
**When** the user types into the search bar or selects a team from the dropdown menu  
**Then** the expense will be filtered for either those that contain the typed in text (case insensitive) in the description, or those that belong to the selected team(s).

## As a user, I should be able to interact with each of my expenses

**Given** the user wants to see more details about each expense in the All Expenses view,  
**When** the user clicks on any expense in the All Expenses list,   
**Then** the Expense Details box will be populated with the following information: if the expense is personal, date, expense amount, description, and personal, and an Edit Expense button is displayed – if the expense is shared, date, total expense amount, current user’s fractional share of the expense, description, team name, name of user who paid, and either amount owed to payee or an “expense settled” success message. If an amount is owed on a shared expense, then a button to Settle Expense will be displayed. If the shared expense was entered by the currently logged in user, an Edit Expense button is displayed.

## As a user who entered an expense, I should be able to edit or delete the expense.

**Given** a user originally entered an expense (the expense.userId matches this user’s user.id),  
**When** the user accesses the Expense Details view for an expense,   
**Then** buttons to Delete Expense and Edit Expense will be displayed. If Delete Expense is clicked, the expense will be removed from the api and the Edit Details section will reset to the default/blank box. If Edit Expense is clicked, the user will be redirected to the Edit Expense view. The Edit Expense view has a pre-populated form with editable input boxes for date, amount description, shared yes/no, and team dropdown, along with a Save Changes button. When Save Changes is clicked, the updated expense data will be put to the api, and the user will be redirected to the All Expenses view.

## As a team member, I should be able to pay the team member who paid the full price of an expense for my share of the expense.

**Given** a currently logged in team member wants to pay back the team member who originally paid for a selected expense,  
**When** the currently logged in team member clicks the Settle Expense button on the Expense Details section of All Expenses view,   
**Then** the user will be redirected to the Settle Single Expense view. In this view, the full amount of the expense, expense description, and date of the expense will be displayed. Below these details, the amount the logged in user owes to the payee will be displayed. Below the amount owed, the user can select one of two radio options: Pay Full Amount, or Enter Custom Amount along with a text input to enter a dollar amount. Below the radios, a Pay Now button is displayed. Upon clicking Pay Now, a new payment is posted to the api, and the user is redirected back to the All Expenses view.

## As a user, I should be able to create a new team.

**Given** a currently logged in user wants to create a new expense-sharing team with other app users,  
**When** the currently logged in user clicks Create New Team on the home page or Your Teams page,  
**Then** the user will be redirected to the Create New Team view. On initial render, this view contains a blank input field for Team Name, a search field to search for other app users, a list of who is currently in the team (on initial render, only the current user is in the list), a blank input field next to each team member where a percentage share can be entered, and a Create Team button. If the Create Team button is clicked, the user will be redirected to the Team Details view for the team that was just created.

## As a user, I should be able to add other app users to a team I am in.

**Given** a currently logged in user wants to add a new team member,  
**When** the user types the username of another app member into the search bar on either the Create New Team or Edit Team views, and a user matching that username exists in the api,  
**Then** the found user’s first name and last name will be displayed, and a button to Add to Team will be displayed. If Add to Team is clicked, the found user will be added to the list of Team Members, along with an empty input field for entering the new user’s percentage share of team expenses.

## As a team member, I should be able to edit my team.

**Given** a currently logged in user is a member of a team,  
**When** the user clicks the Edit Team button from the Team Details view,  
**Then** the user will be redirected to the Edit Team view, where they can update team members’ expense percentages, remove team members, or search for and add new team members. If the Save Changes button at the bottom of this view is clicked, the updated team details will be put to the api, and the user will be redirected to the Team Details view.

## As a user, I should be able to see a list of all the teams I am in.

**Given** a currently logged in user wants to view their teams,  
**When** the user clicks Your Teams in the nav bar or View Your Teams on the home page,  
**Then** the user will be redirected to the Your Teams view. This view displays a list of all teams that the currently logged in user is in. Each team list item includes team name, number of members in the team, and number of shared expenses made by the team to date. There is also a button at the bottom of the Your Teams view to create a new team. If any of the list items are clicked, the user will be redirected to the Team Details view for that team.

## As a team member, I should be able to see details about each team that I am in.

**Given** a team member wants to view details about a team they are in,  
**When** the user clicks the team list item from the Your Teams view,  
**Then** the user will be redirected to the Team Details view. This view displays a list of all users currently in the team and each team member’s percentage share of expenses, the top three spending categories used by this team, and a section for net unsettled debts to/from each team member. In the team members section, each team member’s username will redirect to that team member’s profile. At the bottom of the team members section, there is an Edit Team button that will redirect the user to the Edit Team view. Within the net unsettled debts section, each user other than the currently logged in user will be listed on the left side, and the currently logged in user will be shown on the right. For each user, the amount owed by the user, to the user, or $0 will display.

## As a logged in user, I should be able to see any other app user’s profile.

**Given** a logged in user wants to see details about another app user and they know the other user’s username,  
**When** the logged in user clicks the other user’s username from the Team Details view, or searches for another user from the Create/Edit Team view or their own View Profile view,  
**Then** the user will be redirected to the other user’s profile, which displays their first and last name, username, city, and state. If the logged in user is on a team with the user whose profile is being viewed, then a list of their teams will be listed on the left 1/3rd of the page and a list of recent shared expenses will be listed on the right 1/3rd of the page. Otherwise, the text “You are not currently sharing expenses with this user” will be displayed in both sections. If the currently logged in user is viewing their own profile, then an Edit Profile button will be displayed. If clicked, the user will be redirected to the Edit Profile view.

## As a user, I should be able to edit my own profile.

**Given** a logged in user wants to update information that is displayed on their profile to other users,  
**When** the logged in user clicks Edit Profile on their View Profile page,  
**Then** the user will be redirected to the Edit Profile view. In this view, a form is pre-populated with the current values for first name, last name, city, and state. After editing any of these values, the user can click Save Changes, which will put the updates to the api and redirect the user back to their View Profile view.

## As a user, I should be able to add custom expense categories.

**Given** a user would like to customize the expense categories that can be assigned to their personal and/or team expenses,  
**When** the user clicks Customize Categories from the home page,   
**Then** the user will be redirected to the Expense Categories view. The left 1/3rd of the page displays a list of default categories that any user can use (e.g. groceries, clothes, rent, mortgage, etc.). The middle 1/3rd of the page will be blank if no categories have been created by the user, or will display a list of user-created categories. Next to each user-created category, buttons to edit and delete are displayed. Clicking edit will make the field editable, and a Save button will be displayed. Clicking Save will put the update to the api. Clicking delete will remove the category from the database. The right 1/3rd of the page displays a form for creating a new expense. Given that a user types a unique category into the input field that does not match (case insensitively) any currently existing category and clicks Create, then a new category will be posted to the api and it will be displayed in the middle 1/3rd of the page. If a blank input or duplicate value is entered, the user will receive an error message.

# STRETCH GOALS: STORIES & CRITERIA

## As a team member, I would like to be able to pay each team member a net balance in order to settle multiple expenses at once.

**Given** a team member wants to pay back another team member for multiple expenses  
**When** the a team member clicks the Pay Now button that displays next to a user within the net unsettled debts section of the Team Details view,  
**Then** the team member will be redirected to the Settle Net Expenses view. In this view, the net amount owed to another team member will be displayed. Two radio button options will be displayed: pay full amount, and enter custom amount along with a text input field for entering a dollar amount. A Pay Now button displays at the bottom of the view. When clicked, a payment is created that settles all currently unsettled debts between the two parties.

## As an app user, I would like to be able to see a visual representation of my personal finance snapshot.

**Given** an app user wants to see basic analytics about their spending habits  
**When** the app user logs in to the home page,  
**Then** the app user will see a dashboard summarizing their spending habits. On the home page, a pie chart representing amount spent in each category year-to-date, a bar chart showing spending in each category month-to-month, and a progress bar showing progress towards a user-defined savings goal will be displayed.

## As an app user, I would like to be able to enter multiple new expenses at once.

**Given** an app user wants to enter a longer list of expenses in a single session rather than entering one expense at a time,  
**When** the app user navigates to the Add Expenses view,  
**Then** the app user will see a blank form with multiple lines of inputs. Each line will have inputs for date, amount, description, a dropdown for category, and a dropdown for team. One Submit Expenses button will display at the bottom of the page. When clicked, all expenses entered on the form will be posted to the database.

## As an app user, I would like to be delighted by the user interface and the experience of navigating the app.

**Given** an app user wants to enjoy interacting with the app,  
**When** the app user looks at and interacts with the app,  
**Then** the app user should find the user interface, color scheme, and layout pleasing to look at. Navigating the app should feel intuitive.