*The Room-mate Assistant*

**Author Name:** *Justin Sanders*

**Professor:**  *Dr. McElfresh*

**Executive Summary:**

Moving in with new people can be tough. It is difficult to keep track of finances, chores, schedules, and contact information when moving in with others. This app will provide users a centralized location to assist with each of those issues. Users will always know who to pay, when to pay, chores they have, each other’s information all from one single login accessible online.

Table of Contents

Introduction 3

Project Background 3

Project Description 3

Proposed Solution 3

Development Approach 3

High Level Plan 3

Project Schedule 4

Work Breakdown Structure 4

Project Calendar 5

Intermediate Milestones 5

Intermediate Milestone #1 5

Intermediate Milestone #2 5

Weekly Updates 5

Weekly Update #1 5

Weekly Update #2 5

Weekly Update #3 5

Weekly Update #4 5

Weekly Update #5 5

Weekly Update #6 5

Weekly Update #7 5

Weekly Update #8 5

Weekly Update #9 5

Project Deliverables 6

Conclusion 6

# Introduction

I recently rented a house with two other people. None of us realized how difficult keeping track of who owes who money is. It is also difficult to keep track of everyone’s schedules. I hope that this project will simplify this process of record keeping and understanding why and when you paid for something.

## Project Background

Each of my roommates are trying to split rent, bills, and groceries fairly. This has been a bit tough with our current system of just texting each other or saving receipts to look at later. Implementing a better system that each of us can access at any point in time could be very useful for our future living together.

During my time at CNU I have worked with a couple of databases in CPSC 440. I have also connected a small local website with a SQL database in PHP in CPSC 360. I think these two classes have influenced my idea of building a website and database to support our financial issues.

## Project Description

The project will be a website for use among roommates for finances, scheduling, information storage. The website should be capable of keeping track of all transactions that each roommate makes. It should be able to inform each roommate of other transactions related to them. This includes money that roommate A may owe roommate B. Access to important information on each roommate should be able to be uploaded and viewed by other roommates. Examples of this are: Each roommate phone number, any emergency numbers, emails to be reached at, etc... A schedule of the ‘house’ should be available for roommates to add their days too if they desire to be seen by all roommates.

A group member will be calendar, chores list and shopping list. The calendar will have standard calendar users. Users can add events, remove events, and set recurring events. The chores list will be available to all users and will give the ability to add, remove, and assign chores to other users. The shopping list will be viewable by all members of the group and will be able to add, remove items from the list. Users can also ‘reset’ the list when the groceries have been purchased.

# Proposed Solution

## Development Approach

I intend to take an agile approach to handling this project. This would provide the best experience for my customers (my roommates). The product could also be significantly improved if I were to have roommates use the app and give feedback on certain things they liked and don’t like. Stories can be created based on feedback given and created around items such as, ‘designing the user table in PostgreSQL.

## High Level Plan

I plan to host the database and front end on Heroku. The ‘hobby plan’ for seven dollars per month will give you a small space to host a PostgreSQL database and any front end you make. I aim to use Vue 3 as a framework for the project because it is a straightforward and useful framework. Also, I have some prior experience using the framework on previous projects. User accounts will be done through okta’s free user accounts.

# Project Schedule

The first step in creating this will take a mixture of every aspect of the project. An initial database should be made (maybe 1 table). Then a basic initial front end should be created. After this I will need to connect both together to make sure that I can access the database from the webpage. After this I will need to connect the Okta accounts and study up a bit on how to get a user primary key fort the database. This also involved being able to push my product to Heroku live.

After this I will need to flesh out the database (probably on paper). Try to account for future issues connecting tables together with user accounts. Then build the skeleton of the database. I will need to make a skeleton webpage and get some basic CRUD operations working. Then I can iteratively improve the webpage and database together until all criteria is met.

## Work Breakdown Structure

Database design could take a couple days to account for future changes that will need to take place. Database implementation should only require a day or so but will probably send me back to the design phase to fix issues I find. Front end mockup could take a week or so due to multiple revisions I will need to make after getting user feedback. I’m unsure about Okta account integration time but I imagine it will take a day or so to work properly with the database. Debugging and testing will take maybe two weeks after the completion of the last few steps. I imagine that user feedback will help speed up the debugging process.

# Project Calendar

# Intermediate Milestones

### Intermediate Milestone #1

* Demonstrate working CRUD operations
* User accounts
* Database working
* Financial section working

#### Intermediate Milestone #2

* Most of project finished (Probably buggy sections):
  + Chores list
  + Calendar
  + Finances
  + Members
  + Email invitations

# Weekly Updates

#### Weekly Update #1

* Database designed (1-3 hour)
* Vue project made (30 minutes)
* Git setup (10 minutes)
* Database implementation (2 hours)
* Potential database redesign (1 hour)

#### Weekly Update #2

* Skeleton front end connected to database (1 hour)
* Connect Okta account (2 hours)
* Basic CRUD implementation on database (2 hours)
* CRUD on frontend (1 hours)

#### Weekly Update #3

* Individual user financial tab displaying (4-6 hours)
  + Transactions made by user
  + Debt owed to other users
  + Ability to add transaction
    - Charging other users
    - Write description of transaction
    - Price of transaction
    - Date purchased
  + Users able to charge other users or the whole group different percentages
* Ability to see details of transaction (1 - 2.5 hours)
  + Date added
  + Price listed
  + Description of transaction

#### Weekly Update #4

* Admin Tab for each user information (1-2 hours)
  + Add personal information for user
  + Secondary contact information
  + Parental contact information
* Group admin tab able to (2.5 hours)
  + Add members to each group
  + Remove members from group

#### Weekly Update #5

(Intermediate Presentation)

* Group information Tab (1 hour)
  + View other group members information
* Group Shopping list (2 -3 hours)
  + Users can add and remove any item
  + Any user can reset the list when purchased

#### Weekly Update #6

* Group Chore list (2-3 hours)
  + View all Chores
  + Assign users chores
* Individual view of user’s chores (1 hour)
  + Ability to check off chore

#### Weekly Update #7

* Group calendar where each user can (2-3 hours) probably take longer
  + Add an event
    - Recurring
    - One-time
  + See other group members events
* Email invitation to new group members (2-3 hours)

#### Weekly Update #8

(Intermediate Presentation)

* Testing and debugging (2-3 hours)
* Prepare presentation (1-2 hours)
* Deliverables (1-4 hours)

#### Weekly Update #9

* Backup time if project falls behind (6 hours)

# Project Deliverables

Documentation:

* Paper database mockups
  + Primary keys, foreign keys, tables, etc.…
* System Vision Document: 1–2-page brief description of the vision for this project/system being built

Technical Deliverables:

* Link to git repository with all completed code for database and website
* Link to URL with actively working project for instructor review
* Instructions/readme document explaining how the instructor can install/test the system that was demonstrated, both from a URL of the active site, as well as if the instructor desires to set it up themselves.

Final Presentation Deliverables:

* Demonstration of the website
* Posterboard

# Conclusion

The goals of the project are to have a versatile application for new roommates together. The app will assist new roommates’ common struggles they may experience after moving in together. Group members will be able to see contact information uploaded by other members. They will see finances of the group, keep track of each group members dept. Groups will have a calendar accessible to all members, a chores list, ability to assign chores to other group members. A shopping list will be viewable by all members of the group and will be able to add, remove items from the list. Users can also ‘reset’ the list when the groceries have been purchased.

The project should take roughly the whole nine weeks to finish thoroughly. The only areas of concern I have are issues with services I plan to use. Issues with things such as user accounts, the database, and Heroku hosting. My only concern is that these are services I will not have direct control over.

List of Features to be delivered

* Individual user accounts
* Individual user financial tab displaying
  + Transactions made by user
  + Debt owed to other users
  + Ability to add transaction
    - Charging other users
    - Write description of transaction
    - Price of transaction
    - Date purchased
* Admin Tab for each user information
  + Add personal information for user
  + Secondary contact information
  + Parental contact information
* Group information Tab
  + View other group members information
* Users able to charge other users or the whole group different percentages
* Group admin tab able to
  + Add members to each group
  + Remove members from group
* Group calendar where each user can
  + Add an event
    - Recurring
    - One-time
  + See other group members events
* Group financial tab for all transactions made by each member
* Ability to see details of transaction
  + Date added
  + Price listed
  + Description of transaction
* Group Shopping list
* Users can add and remove any item
* Any user can reset the list when purchased
* Group Chore list
  + View all Chores
  + Assign users chores
* Individual view of user’s chores
  + Ability to check off chore

Potential Features (A finished application)

* QR code that allows users to easily join a new group
* Email invitation to new group
* Venmo/PayPal integration with user account
* SMS/Email notifications on calendar events
* Separate phone application with similar functionality as website