Homework - Week 2 Software
Group 1 - Heidi, Sarah, Emily, Myrtle, Rachel

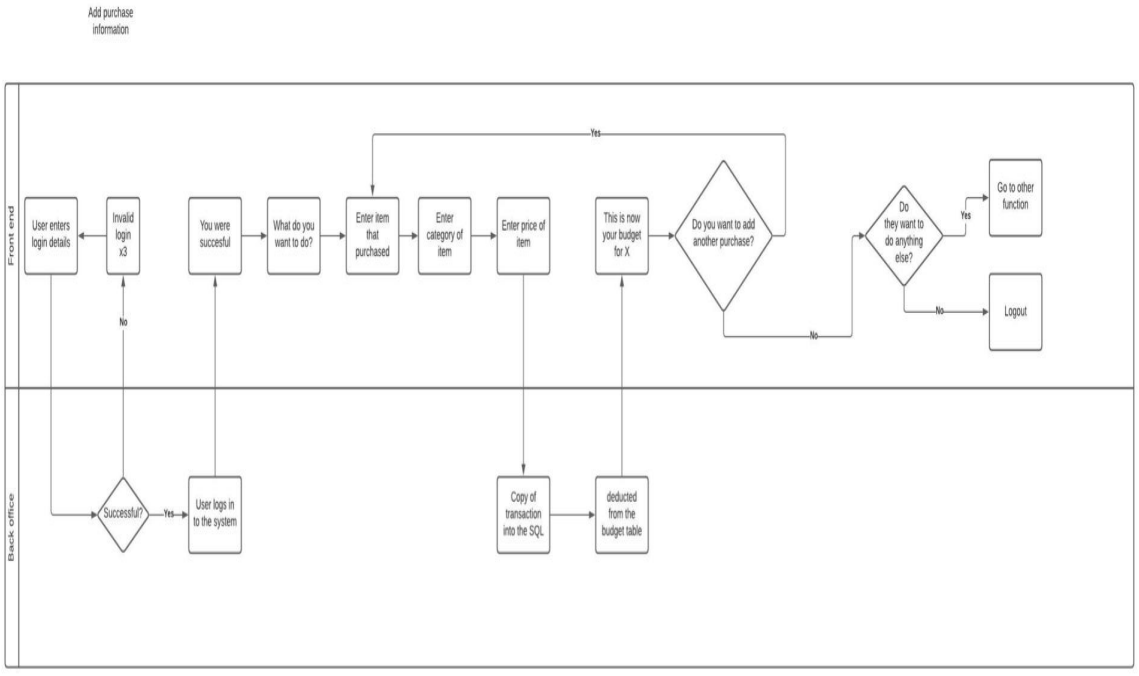
We have decided to build a fun and motivational budgeting application which will enable our users to view and edit their monthly budget, to enter purchases and set savings goals. As a group we agreed that budgeting can be confusing and inaccessible. We want to make it simple and straightforward for anyone who hasn't used a budgeting app before. We also considered that using music to motivate and cajole users into reaching their savings goals and staying on track with their spending could be an excellent introduction to managing money for young people and encourage them to take control of their finances from an early age.

These will be the key features of our application:

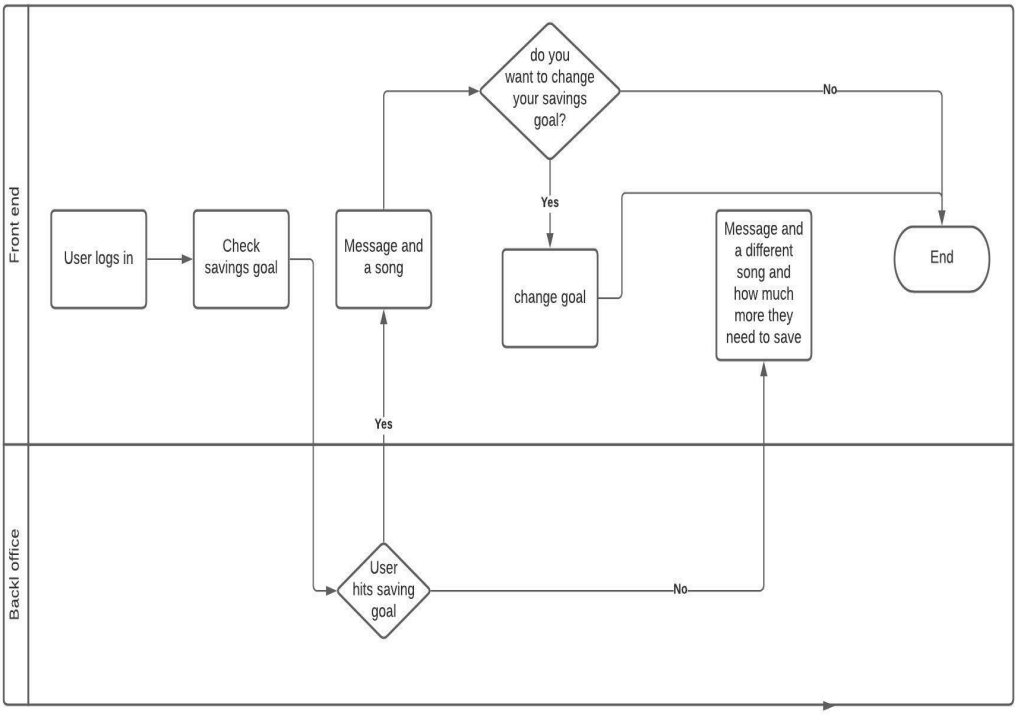
- Enables the user to set budgeting and saving goals
- Allows the user to enter and track their spending on items by category
- Provides the user with congratulatory music once a saving goal is reached. E.g. the Gold Diggers Song (We're in the Money)
- Motivational music will be played to the user if their saving goal is yet to be reached. E.g. Nine to Five by Dolly Parton

We have created three user journeys below:

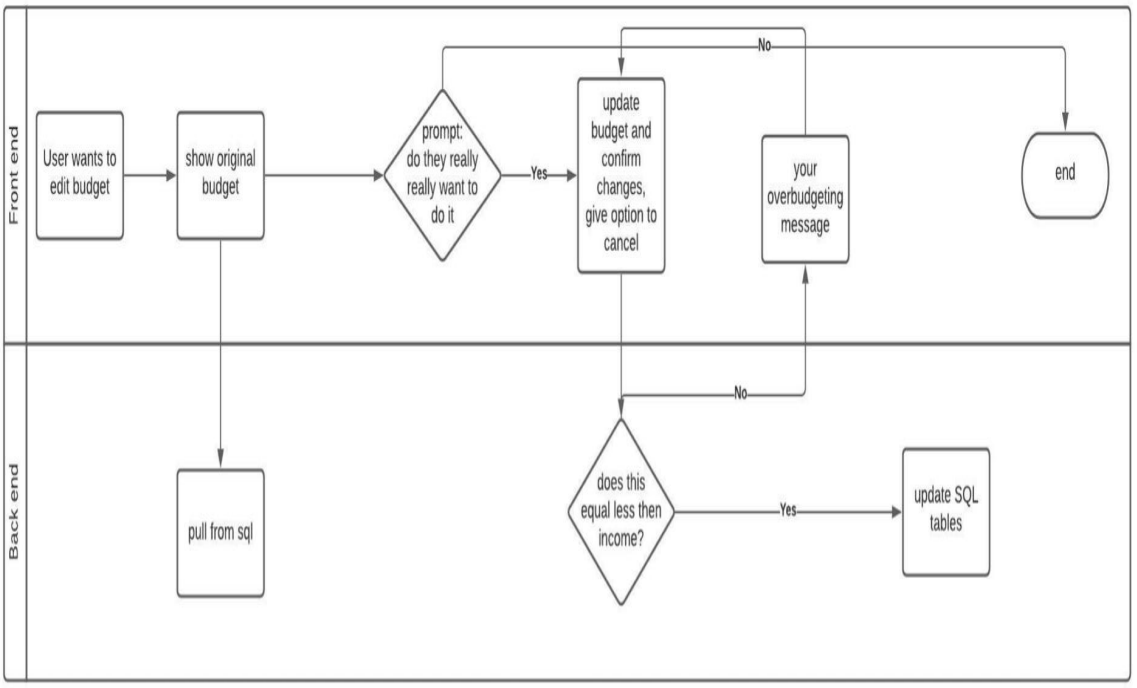
User Journey 1



User Journey 2

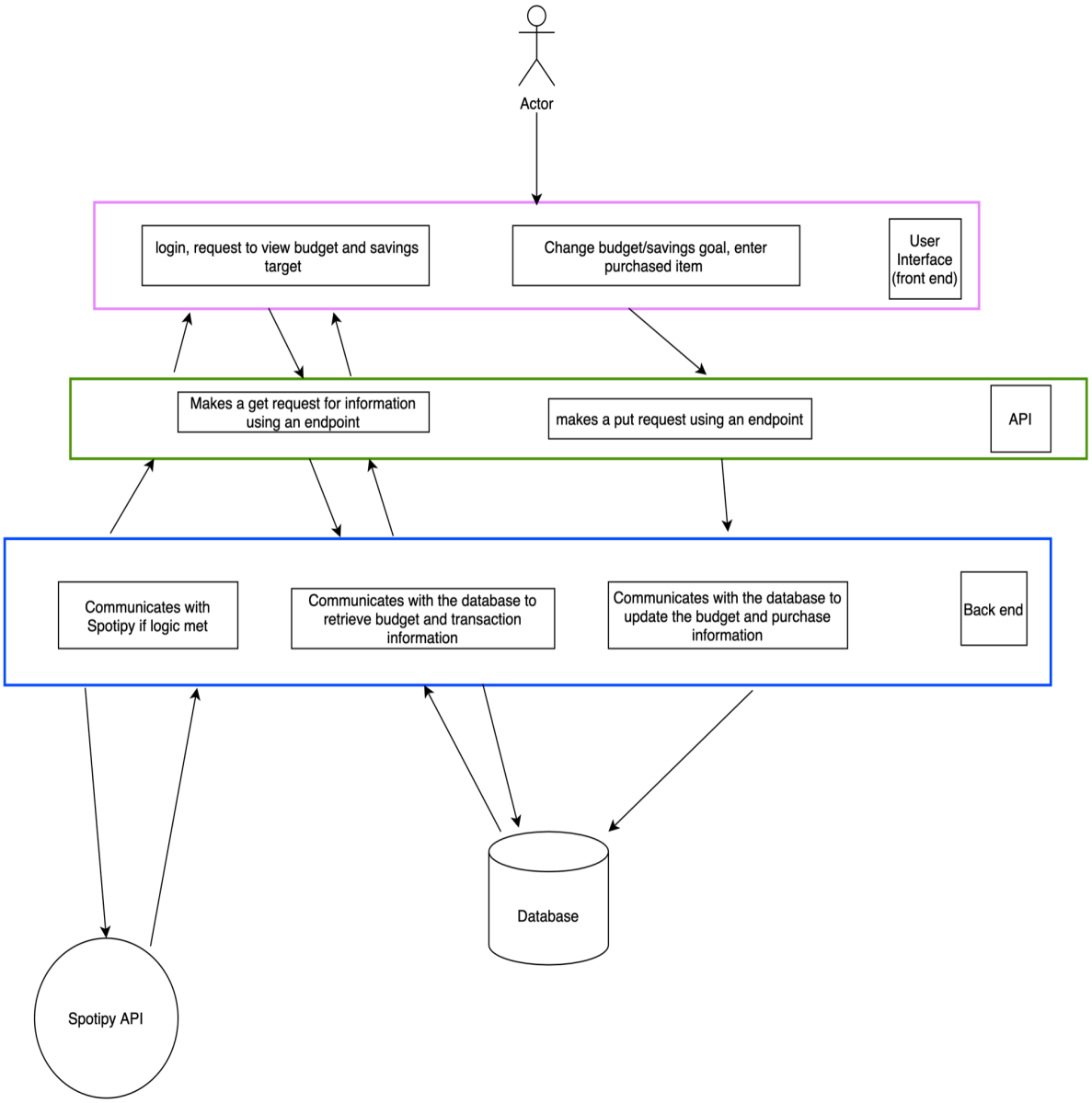


User Journey 3



System Architecture

This will consist of a database with user information, budget and transaction data. Our backend logic will communicate with the database and the Spotify API and the front end will request user input and return data from the database using Flask to create an API.



Team approach

Initially we met via zoom and worked together to share ideas about which features the app should have. Once initial requirements were finalised we prioritised the tasks which will need to be completed first. We believe that the database and API need to be set up fairly quickly in order that the logic and functions can be tested. Once we have these set up and have a basic working product we will refine and improve as we go. In this way we are following an Agile model of software development rather than the Waterfall method.

In terms of allocation of workload Sarah and Rachel are working concurrently to create the database using MySQL and to create the API. Heidi, Emily and Myrtle will code the back-end logic using Python. A plan of the functions we think will be needed at this stage has been created and we are mindful of the need to refactor these into classes with attributes and methods appropriate for the application.

We plan to test our application continually throughout the software development cycle, testing against our requirements and also using family members to trial it and provide feedback. We are considering a test script but also aware of the benefit of allowing users to ‘free-test’ it to request data or produce results we haven’t thought of ourselves e.g edge case scenarios we may not have predicted.