Sean Morrison, Adam Todd, Joshua Brunscheen, Joshua Javillo, David Wang

Dr. Callahan

Web Science Systems Development

January 30, 2024

Budget Buddy

The idea:

Budget Buddy is meant to be an application meant to assist smaller organizations or individuals who don't have a lot of time or knowledge to be effectively budgeting. The application is meant to rival other premium budgeting applications options that require a subscription or one time payment fee. It also aims to rival traditional methods of tracking expenses such as using excel or simply saving physical receipts. Budget Buddy would offer a free and easy to use service for smaller organizations like clubs or start up businesses. Simultaneously, Budget Buddy can be used by individuals of any age to track their own personal finances. A feature that makes our service unique is the ability to use our "AI receipt scanning technology" for users to either upload or take pictures of their receipts and then effortlessly record the data into whatever budget plan they want.

The stakeholders:

Our stakeholders are derived from 3 groups: Small organizations, small businesses, and the individual user. Budget Buddy will help both small organizations and businesses to help track expenses and plan for financial goals. In terms of differences between the organization and the business, our app will provide resources that are tailored to the users needs. Budget buddy can generate expense tracking reports for organizations and help manage bills for businesses. As for the individual, our app will generate good practices and a good financial understanding for those

who are becoming adults–namely, the college student. Another benefit to our stakeholders is our cost effective design model. Our app will be completely free and hence will not only help users save money, but they will also be saving money just by using the app.

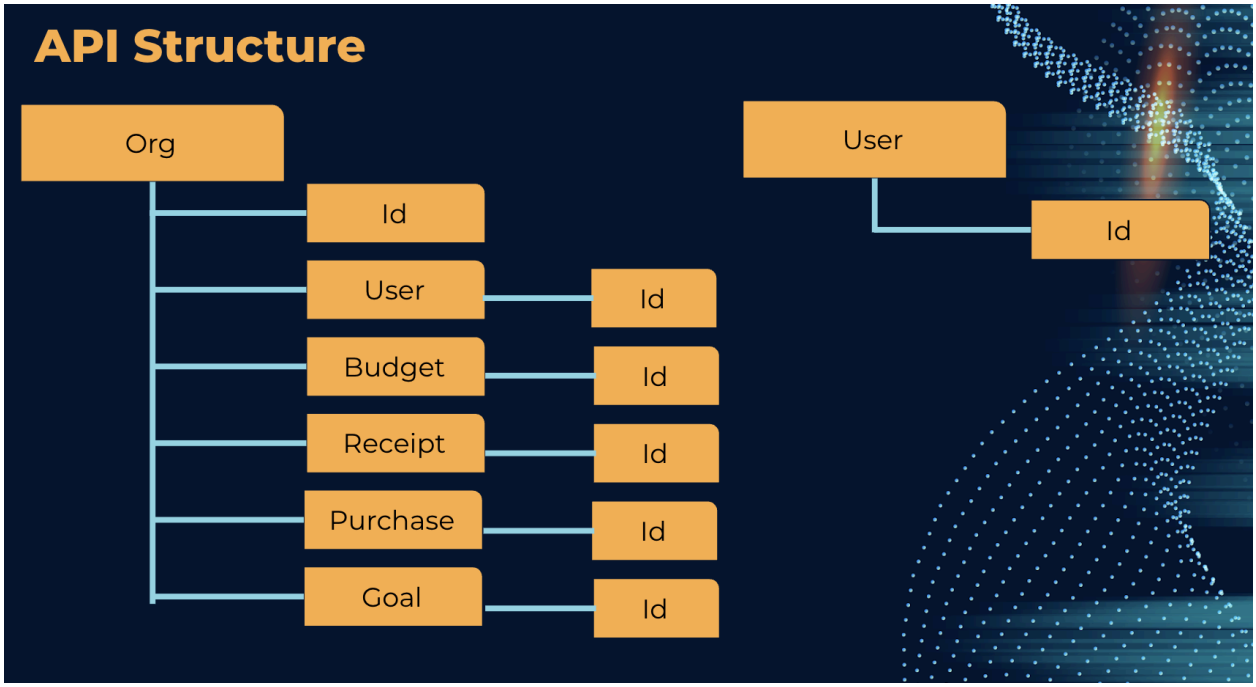
The tech stack: (functional/non functional requirements, api endpoints)

In terms of our tech stack, we will be utilizing the component based features of React.JS to create an efficient and dynamic web app. We will apply TailwindCSS to create a visually compelling front end taking into account optimization features. Our database will use MongoDB to store our user's data. We will be obfuscating this data between our server and our front end to ensure user data is securely transferred and security is maximized.

Functional/Non-Functional Requirements:

We have the following functional requirements: a database, appealing front end, receipt scanning capabilities, and financial analytics. Our database will fulfill a variety of purposes such as keeping track of users, organizations, and individual information for each of those groups. It will also keep track of permissions between these entities. For example, a user who is not the owner of an organization should not be able to delete the organization. This database also has to keep track of receipt image uploads for organizations. The goal of our frontend is to make it more appealing and simpler than Microsoft Excel; we want our application to be preferred by the general public in terms of ease of use and aesthetics. Of course, in order to scan receipts, our application has to allow image uploads and AI technology. We have not decided on a library or third party application for scanning the receipts but we have a few ideas. One example is the JavaScript JS Receipt OCR. Finally, our application must use a variety of statistical methods to organize an organization's financial data. At the very least, we want to calculate how much of the budget an organization has used along with calculating the progress on an organization's goals.

We also have a few non-functional requirements. If this wasn't a school project, we would have liked to implement bank integration using an application such as Stripe. However, we obviously cannot do this for a school project for security and privacy reasons. Another functional requirement would be to give general financial and stock advice to users to better manage their money. However, we would need more qualifications to give financial advice. Finally, a fully functional Budget Buddy could scan the Internet for potential coupons for the organization to maximize savings. Our application would do this by analyzing the organization's shopping trends (with the organization's permission) and looking up deals and coupons based on those trends.



Planned API endpoints:

endpoint	Get	Post	Put	Delete
/org	IDs of orgs	Create new org	Null	Delete every org (for devs only)
/org/id	Get org info	null	Update org info	Delete that org
/org/user	All users of org	Add user to org	Mass update	Delete all users
/org/user/id	Get user info	null	Update this user	Delete user
/org/budget	All budgets of org	Add budget to org	Mass update	Delete all budget
/org/budget/id	Get budget info	null	Update this budget	Delete budget
/org/receipt	All receipts of org	Add receipt to org	Mass update	Delete all receipt
/org/receipt/id	Get receipt info	null	Update this receipt	Delete this receipt
/org/purchase	All purchases of org	Add purchase to org	Mass update	Delete all purchases
/org/purchase/id	Get purchase info	null	Update this purchase	Delete this purchase
/org/goal	All goals	Add goal to org	Mass update	Delete all goals
/org/goal/id	Get goal info	null	Update this goal	Delete this goal
/user	Get all user IDs	Add new user	Update all users	Delete all users
/user/id	Get user info	null	Update user info	Delete user

Intended Timeline

February

- Get a desktop frontend
- Get started on the back end

March

- Mid term presentation
- Finish backend
- Implement basic scanning feature

April

- Mobile
- Debug