Team Name: Team 0

Team Member and Emails:

* Daniel Kwong, dkwon014@ucr.edu
* Rahul Nair, rnair005@ucr.edu
* Sandy Tsan, stsan003@ucr.edu
* Tyler Ortiz, torti006@ucr.edu
* Jeremy Taraba, jtara006@ucr.edu

Project: Budget Managing

Communication Channel: Facebook Messenger

Tech Presentation Topic : Django

Features

1. User account

As an interested user of the app, I want to be able to register and log in to the app.

1. Link banking account to the app

As a user who wants to manage their finances from one app, I want to be able to connect my bank account to the app so I can view my spendings and current amount from it.

* 1. Display current amount in account
  2. Display spendings within that account, options for daily/weekly/monthly spending

1. Push notifications of spending

As a registered user, I want to be notified of my progress in spending and saving money.

* 1. Text notification on phone about spending too much?
  2. “You spent $10 less than you did yesterday, good job!”
  3. “You spent $100 yesterday… don’t buy anything today”

1. Account based on google accounts

As an interested user, I want to be able to quickly sign in using my Google Account instead of using the routine sign in.

1. Security authentication!!! CAPTCHA or something before logging in

As a registered user, I want to know that my data is secured and only I can access it.

1. Search bar to search spendings → search for spendings within categories

As a user who buys a lot of stuff, I would like to be able to find the price of something I bought last week

1. Export/Import spendings in .csv file

As a user who has been keeping their spendings in an excel sheet, I would like to easily transfer my spending data into the app in 1 click

1. Data analytics of your spendings for x period of time

As a user of the web app, I want to be able to see my spendings through organized, aesthetic graphs and analytics

1. Savings/Debt calculator

As a user of the web app, I want to be able to accurately calculate how much I need to save without having to go through the pain of calculating everything myself

* 1. If I spent x amount of money each day while gaining x amount for each day, I would earn x amount after x days

1. In App calculator

As a user of the web app, I want to do all of my work in one space and not have to switch between two apps

1. MFA like inputting a pincode before accessing the budgeting app

As a registered user, I want my account to be secure and validate every log in was performed by me.

1. Suggested amount to put into savings based on spending habits

As a registered user, I would like to receive recommendations on saving money based on my current habits.

1. Suggested amount to put towards debt i.e. student loans, credit card, car payment, mortgage

As a user in debt, I would like to be able to see how long it is going to take to pay off my debts.

1. Being able to set goals → want to save x amount of money

As a prodigal user, I would like to set an amount of money to still have in my bank account when the week/month is over

1. Being able to set a budget for the month

As a registered user, I would like to set a budget for a month and not be allowed to go past it.

1. Track all bills and set reminders for bills

As a forgetful user, I would like to know when all my bills are due and have reminders notifications for them

1. Show exchange rate for other currencies

As a foreign user, I would like to know up-to-date exchange rates for my country.

1. Set spending warnings and limits to follow certain budges

As an interested user, I want to be able to set boundaries and limitations on my spending.

1. Spending categories shown on the web page

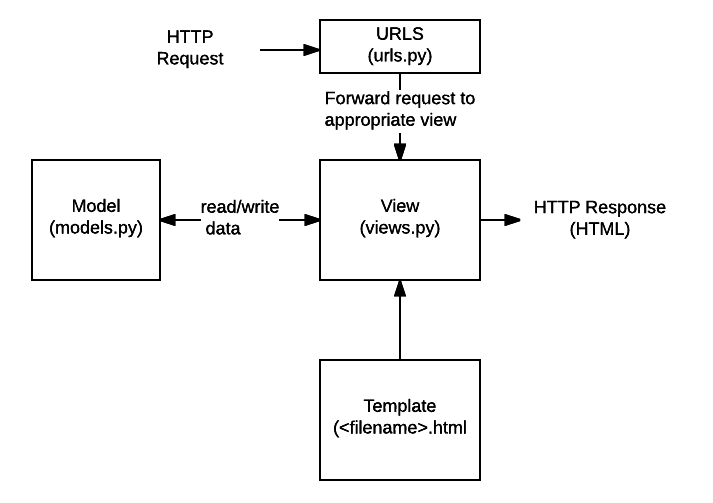
As a registered user, I would like to see the different categories of my spending on the

website.

\*\*\*\*\*TOPIC FOR PRESENTATION\*\*\*\*\*\*\*\*  
Presentation slides [here](https://docs.google.com/presentation/d/1VCdNsyfoYUPn6ig9ndzRf7af7DnpvFiTSbBPZK-eqgk/edit?usp=sharing) or shared to your rmail.

**DJANGO**

* MVC core architecture (**M**odel **V**iew **C**ontroller)
* Files and data models are written in Python which is used throughout the framework
* Security features to prevent clickjacking, cross-site scripting, and SQL injection  
  Clickjacking - where a malicious site wraps itself in frame, Cross Site Scripting (XSS) is where a user is able to inject malicious client side scripts into the browser of other users, SQL injection is an attack where a user can execute SQL code to delete or leak a database
* Used to manage Instagram, Youtube, Dropbox, Spotify, etc.
* Scalable
* Django is unique as it has three types of files that handle website requests also known as a model-template-view architectural pattern, picture here → <https://www.tutorialspoint.com/django/django_overview.htm>. The developer provides the model, view, and template and links it to a URL that Django will use to present to a user.
  + URLs (py) → Instead of processing URLs with one big function, Django processes URL redirects and sorts them into different viewing functions, which is more maintainable.
  + Models (py) → Receive viewing data and compile it into a comprehensible structure. Allow for database management and query.
  + View (py) → Views in django are request functions. They receive requests and return responses. The requests are received from the URLs and the responses are processed through models
  + Template (html) → Templates are just what it is… templates. They have the structure of the website and output whatever is programmed to them





Piggy Bank (Web Version)

Description: The software project is a web based implementation of a budgeting app. It will allow users to view their spending habits while giving suggestions on how to better spend and budget their hard earned money. Our focus is to streamline the budgeting and saving process as much as possible, from the minimalistic UI to the easy budgeting calculator.

User Stories:

* **User account and authentication -** As an interested user of the app, I want to be able to register and log in to the app.
* **Link banking account to the app -** As a user who wants to manage their finances from one app, I want to be able to connect my bank account to the app so I can view my spendings and current amount from it.
* **Display current amount in account** - Display spendings within that account, options for daily/weekly/monthly spending
* **Push notifications of spending** - As a registered user, I want to be notified of my progress in spending and saving money.
* **Account based on google accounts -** As an interested user, I want to be able to quickly sign in using my Google Account instead of using the routine sign in.
* **Security authentication** - As a registered user, I want to know that my data is secured and only I can access it.
* **Data analytics -** As a user of the web app, I want to be able to see my spendings through organized, aesthetic graphs and analytics
* **Savings/Debt calculator** - As a user of the web app, I want to be able to accurately calculate how much I need to save without having to go through the pain of calculating everything myself
* **In App calculator -** As a user of the web app, I want to do all of my work in one space and not have to switch between two apps
* **Suggested amount to put towards debt i.e. student loans, credit card, car payment, mortgage -** As a user in debt, I would like to be able to see how long it is going to take to pay off my debts.
* **Show exchange rate for other currencies -** As a foreign user, I would like to know up-to-date exchange rates for my country.
* **Spending categories shown on the web page -** As a registered user, I would like to see the different categories of my spending on the website.

Design Document

App Architecture

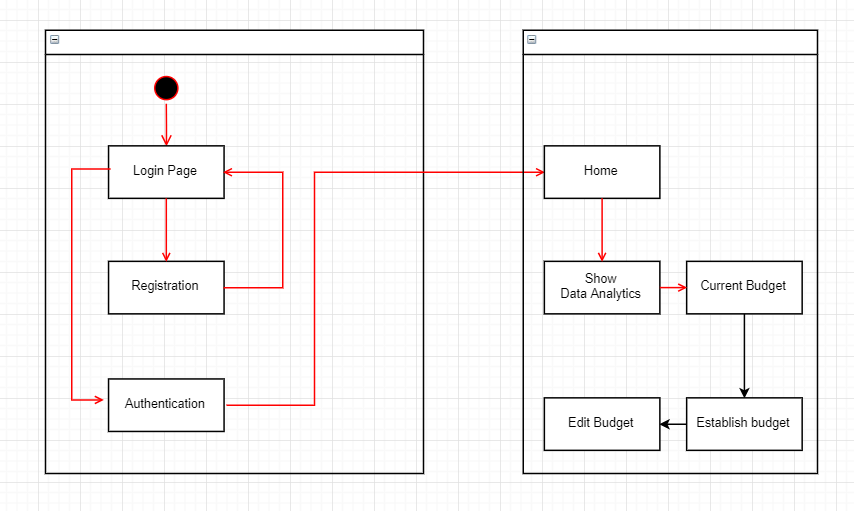
Web-based application using React and Firebase.

Data Model

User accounts will be stored in a database attached to their information such as money in accounts and the different spending categories.

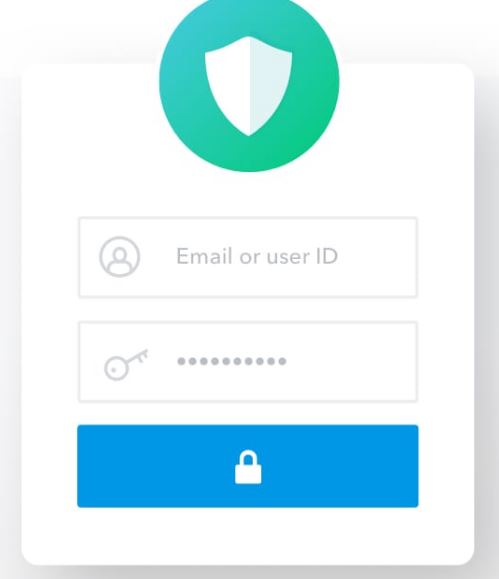
UML Chart

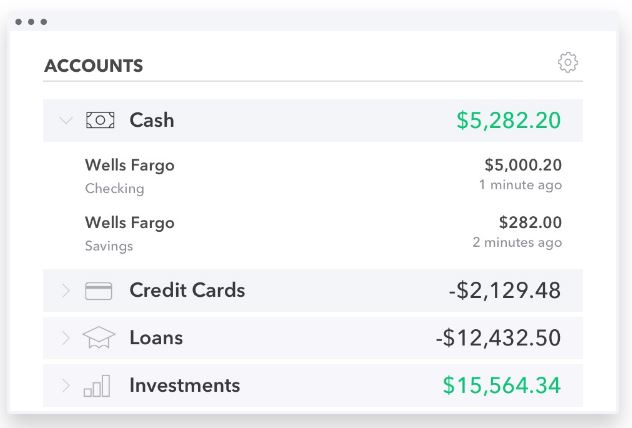
User first encounters login page before going into the home page



UI:

**UI** is designed in such a way that related things are combined or close to each other and unrelated things are separated. In the case of a mistake, the system displays an error message. Through a system of messages, the design keeps users informed of actions, errors, or other feedback. We use the same names to perform the same operations with different objects in order to reduce ambiguity and redundancy.

**Login/Sign-up Page:** Description of what Piggy Bank does and under this the user is prompted to enter login credentials or sign-up for a new account

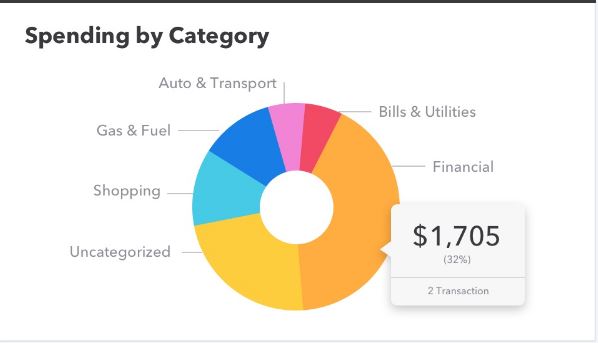


**Home Page:** Displays current balance of accounts as well as bills and monthly spending with a tab to access other options

**Savings/Debt calculator:** Calculates how much you can spend or need to save to reach a certain goal

**In-App Calculator:** functions like a normal calculator

**Expansion of Monthly/Yearly Spending:** shows a list of spending habits, dates/time, prices, categorized



**Exchange Rates:** Works like google translate, input source currency and amount and choose the output currency.

Function Descriptions:

Login

**Registration** - user enters email and password as well as confirms their password and captcha. Optional send authentication email

Authentication

Home

Show Data Analytics

Current Budget

Establish Budget

Edit Budget