

# Design Doc: Team 2003 Honda Civic

Jack Schluger, Sebastian Cain, Noah Fichter, Chloe Delfau, Asher Lasday

Yabe, so named as the reverse of eBay, is ... the reverse of eBay. Instead of users listing things they wish to sell for buyers to bid higher and higher on, users will list things they want to buy, and sellers will have bidding wars for the lowest price. Users will create accounts to buy and sell items, and sales will be made through PayPal, using the PayPal API.

## Roles:

Noah - API Management

Jack - Database Management

Chloe - app.py and backend

Sebastian - frontend

Asher - Project Manager

## Database Schema:

We will have one database: database.db. In this database.bd, there will be a few tables:

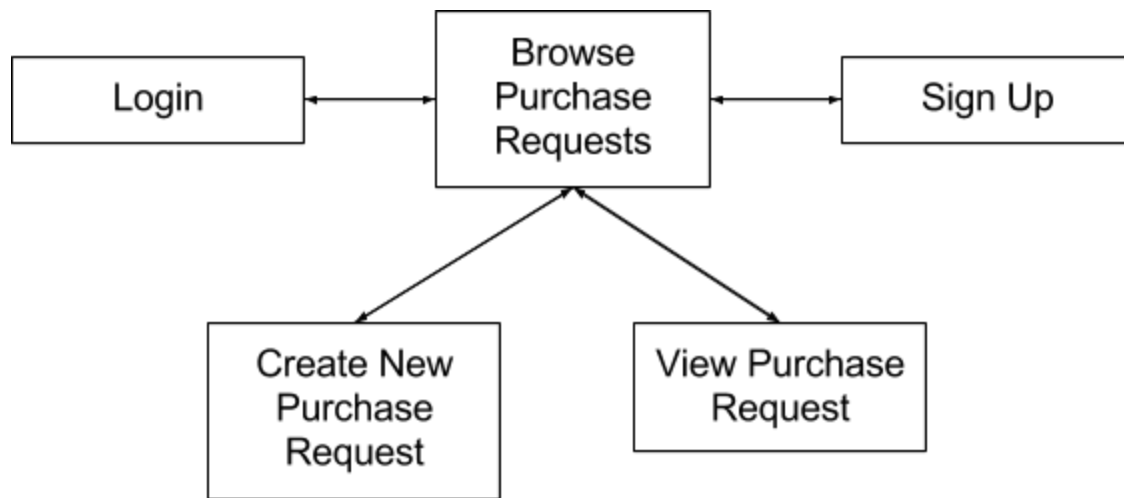
- users
  - username TEXT
  - password TEXT
  - userId INTEGER
  - email TEXT
  - addr1 TEXT
  - addr2 TEXT
  - addrCity TEXT
  - addrState TEXT
  - addrZip TEXT
  - nameF TEXT
  - nameL TEXT
  - phone TEXT
  -

- posts
  - owner TEXT
  - postId INTEGER
  - title TEXT
  - startingPrice INTEGER
  - date FLOAT
  - expires FLOAT
  - active INTEGER
  - imageLink TEXT
  - description TEXT
- bids
  - bidder TEXT
  - postId INTEGER
  - bidId INTEGER
  - price INTEGER
  - date FLOAT
- sales
  - postId INTEGER
  - bidId INTEGER
  - saleId INTEGER
  - date FLOAT

I will use the file `utils/dbManager.py` to interact with the database. There will be the following methods in this file:

- `update_posts()`
- `new_post(owner, title, startingPrice, period)`
- `get_post(postId)`
- `get_posts(number)`
- `get_posts_by_username(username)`
- `new_bid(bidder, postId, price)`
- `get_bid(bidId)`
- `get_bids(postId)`
- `lowest_bid(postId)`
- `log_sale(postId, bidId)`
- `end_post(postId)`

## Site Map:



## API Information:

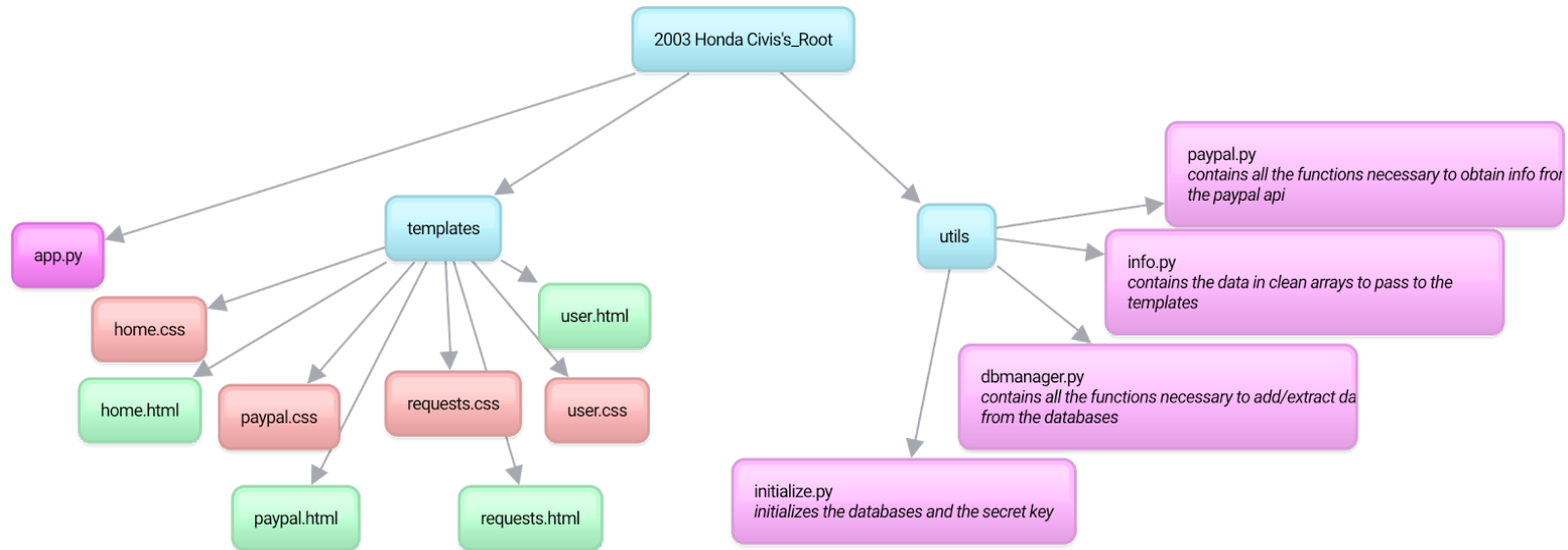
APIs to be used: PayPal

PayPal: used to create an invoice from the seller to the buyer

Functions: create invoice, check if invoice has been paid, create shipping label (extra feature)

API calls will be done with requests library

# File Tree



created with [www.bubbl.us](http://www.bubbl.us)

## Examples of potential routes:

- Home page: `@app.route("/")`
  - Includes a login and registration form
  - These forms will be where a user is redirected to if (s)he tries to buy/sell something without being logged in
- User account: `@app.route("/user")`
  - Shows all account information
    - Items bought
    - Items sold
    - Buy requests
    - Sees items you've saved (potential save for later option)
- Buy requests: `@app.route("/requests")`
  - Shows all posts of requests people have made and prices they're willing to pay
- Paypal: `@app.route("/paypal")`
  - Shows all relevant information about your paypal account
    - All your transactions
    - How much money is on your account