Ibrahim Al-Saud
Emanuelle Crespi
11/17/2016 - CMSC424

Phase 2 report (Updates to Phase 1 **)

Purpose

Our purpose in this phase lies mostly in organizing our data dependencies into a normalized form from the logical E-R model we have discussed in class, and conceptualizing the tasks for scraping/loading our database. We have also dedicated this phase to demoing/testing a client-server interaction, confirming the connection between MySQL and Apache. The server is responsible for receiving user requests, in select query form, and will request from the database; thus, giving the server a temporary relation that is turned into an html page to be displayed to the user.

Data Sources **

Elections data by year
- http://www.270towin.com/historical-presidential-elections/

Percentages/Ballot Statistics
- https://www.loc.gov/rr/program/bib/elections/statistics.html
- http://www.infoplease.com/ipa/A0781450.html

U.S. population
- https://fusiontables.google.com/DataSource?dsrcid=225439#rows:id=1

Candidate Homestates
- https://en.wikipedia.org/wiki/List_of_Presidents_of_the_United_States_by_home_state

Political Parties
- https://en.wikipedia.org/wiki/List_of_political_parties_in_the_United_States
- http://www.globalelectionsdatabase.com/index.php/index

List of Candidates
- https://en.wikipedia.org/wiki/List_of_United_States_presidential_candidates

Campaign slogan (1840 - 2016)
- https://en.wikipedia.org/wiki/List_of_U.S._presidential_campaign_slogans
- http://www.presidentsusa.net/campaignslogans.html

Campaign Expenses
- http://www.fec.gov/disclosurep/pnational.do

Net Worth of candidates
- http://www.huffingtonpost.com/2011/02/21/the-net-worth-of-the-amer_n_825939.html
- http://www.usatoday.com/story/news/politics/elections/2015/08/26/24-7-wall-st-net-worth-presidential-candidates/32409491/

Assumptions and Limitation

- User speaks/reads English language.
- Understands US election process.
- Know basic commands for SQL queries.
- User does not need authentication to access.
- Basic web browsing skills.

The top-level information flow diagram **

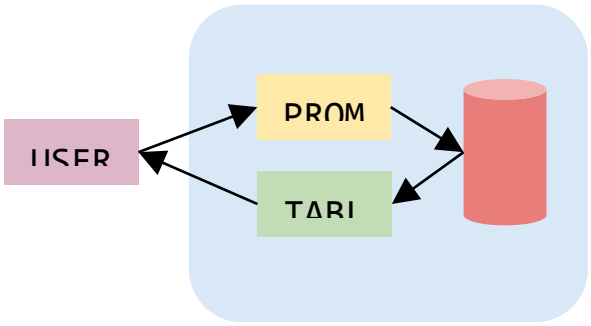

Figure 1: User Perspective (NOT SURE HOW TO FIX THIS??)

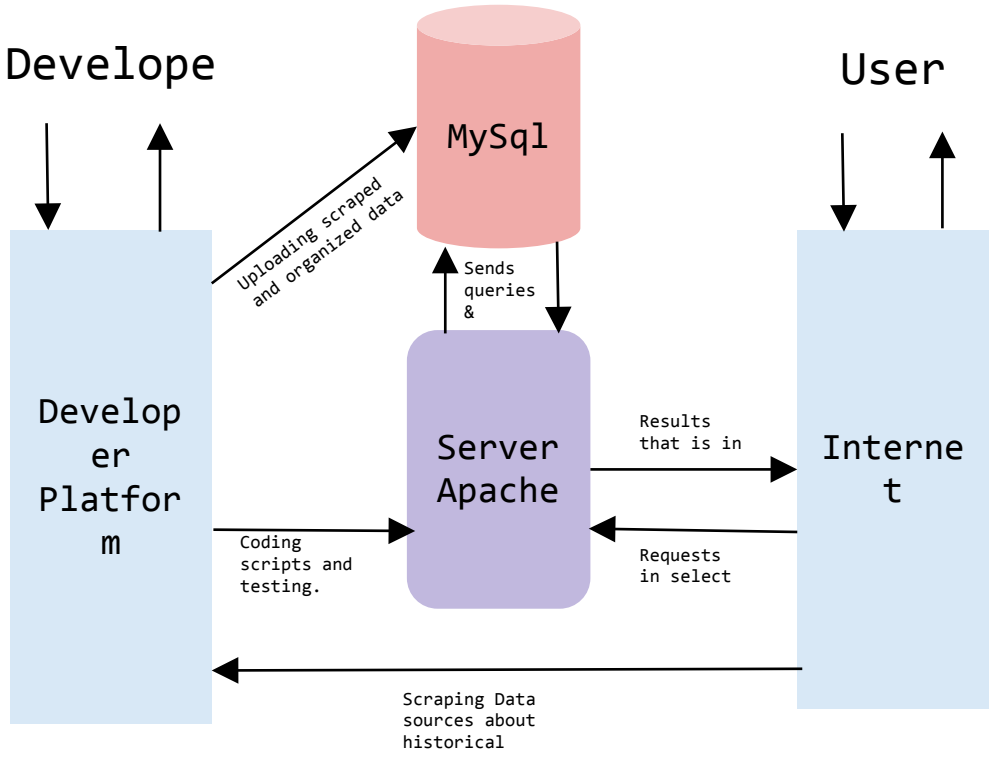

Figure 2: Developer Perspective

The data documents that carry data between tasks **

Data from web sources are going to be scraped using python, version 2.7, and crawling libraries such as Scrapy and Selenium. In addition, html trees of those data sites will be analysed using tools such as BeautifulSoup. Data carried between each task will be in standard JSON format. JSON files will be created and saved as backup.

After the crawling, the program responsible for uploading to the database will parse JSON files and make sure that the data is cleaned and is fit for database entry. The data will be entered to the election database using MySQLdb library, allowing us to execute queries using python.

MySQLdb: https://pypi.python.org/pypi/MySQL-python/
Scrapy: https://scrapy.org/
Selenium: http://docs.seleniumhq.org/
BuetifulSoup: https://www.crummy.com/software/BeautifulSoup/

Progress Report on DBMS System

We have a virtual host connection setup with Apache via XAMPP (Figure 3). Instead of using the php, we have a python script to query our database as we receive requests from the user. The script executes the corresponding cgi file to display the table dynamically from the user's query (Figure 4). We plan to provide the user with the logical model and the relational schema as a convenience for fast and easy querying. We will also modify the query output to a more readable/presentable table format.

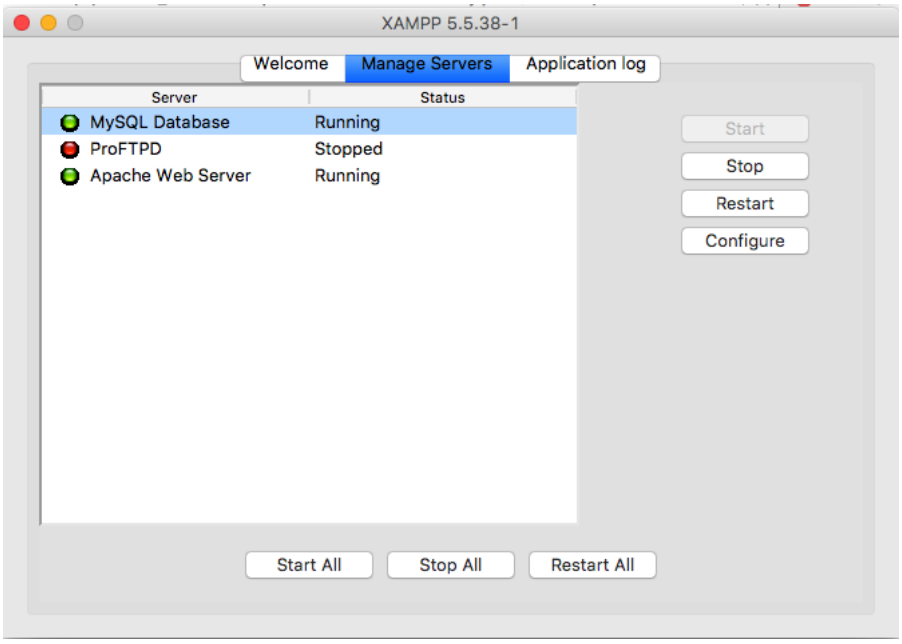

Figure 3: XAMPP (Apache Web Server Enabled)

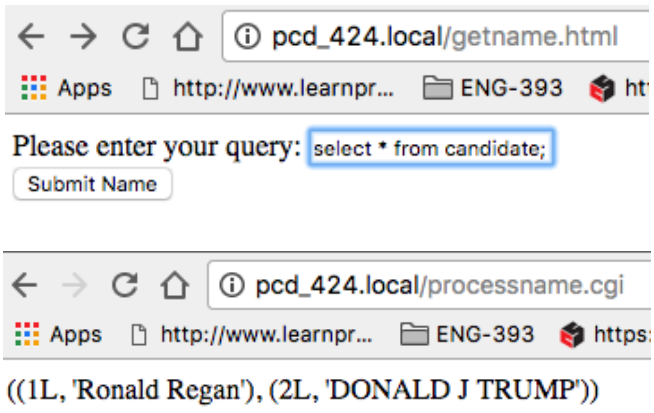

Figure 4: Example User Interface Resulting in Table from Query

Pseudo-Code Describing DML Process

After scraping the data and organizing into JSON files, a program responsible for uploading data will be run. It will parse in the JSON files and cleanse the data. It will make sure that the data is in a format that agrees with the database attribute types. When all of these checks have been crossed, we begin uploading the data using MySQLdb using python 2.7. We then plan to make our final modifications to the overall structure and layout of the html page for phase 3.
[*Please refer to UML (Figure 5) and logical model (Figure 6) on the last 2 pages*]

Scraping for An Arbitrary Table

```
# if website is javascript website, must wait to load
HTML_tree ← selenium to load the static webpage

# using beautiful soup to parse HTML info
dictionary = beautifulsoup.parse( HTML_tree )

# generate JSON file
jsonfile = json.parse(dictionary)
File = open("output.json")
File.write(jsonfile.string())
```

Example Demonstrates Uploading for Nominee Table

```
# parse in the json corresponding nominee json file.
File = open(output.json)
jsondictionary = json.load(file.readall)
dictionary = jsondictionary.dictionary()

# clean data in dictionary
# align data with database types
# fail if not satisfied.

# Connect to the database
db = MySQLdb.connect(host,user,passwd,dbName)
cur = db.cursor()

#create queries
For each DML row in dictionary:
        Query = “insert”+”....”
        cur.execute(query)

# uploads and executes the queries to and on the database.
db.commit()
```

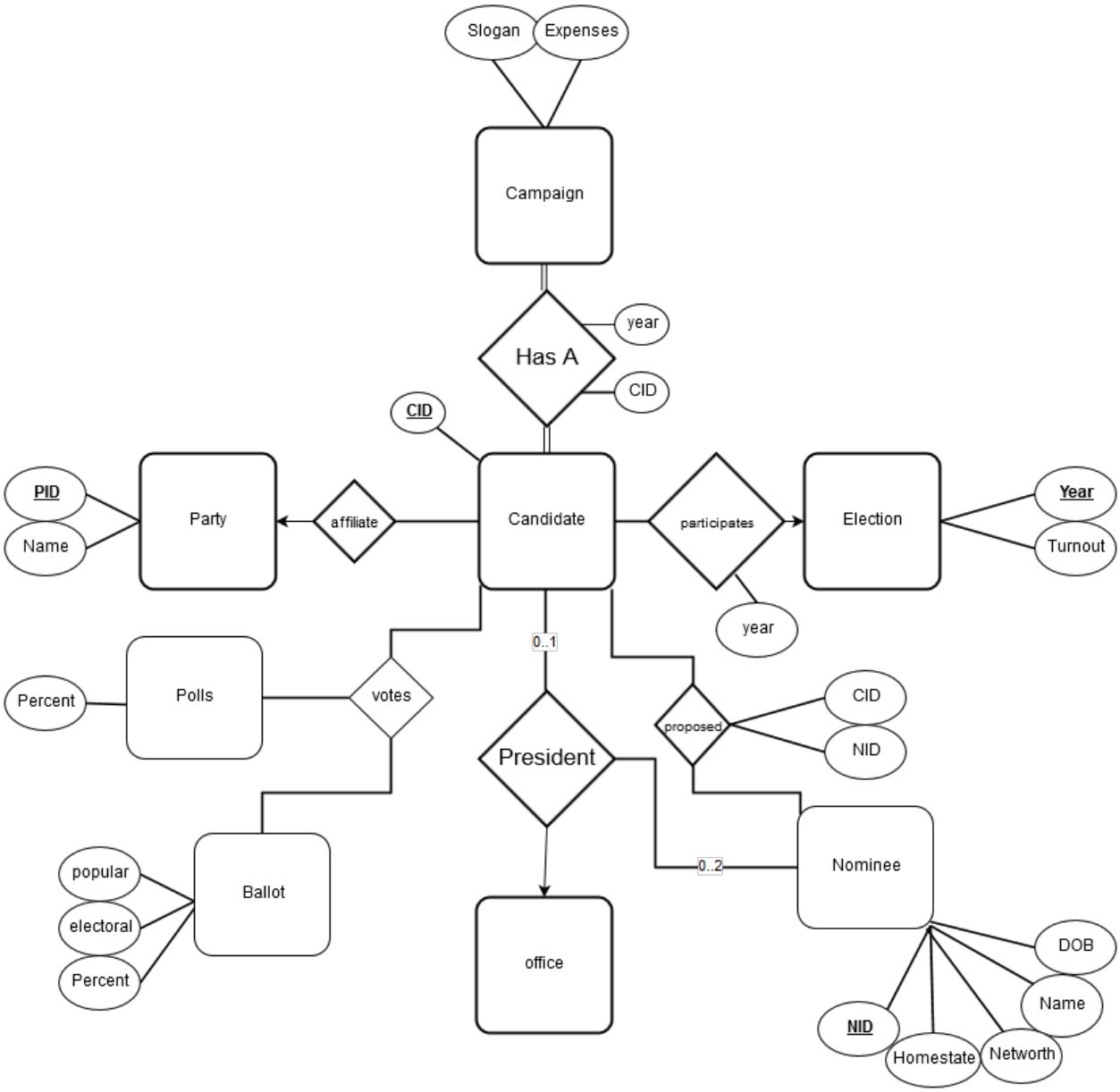

Figure 5: Logical Model

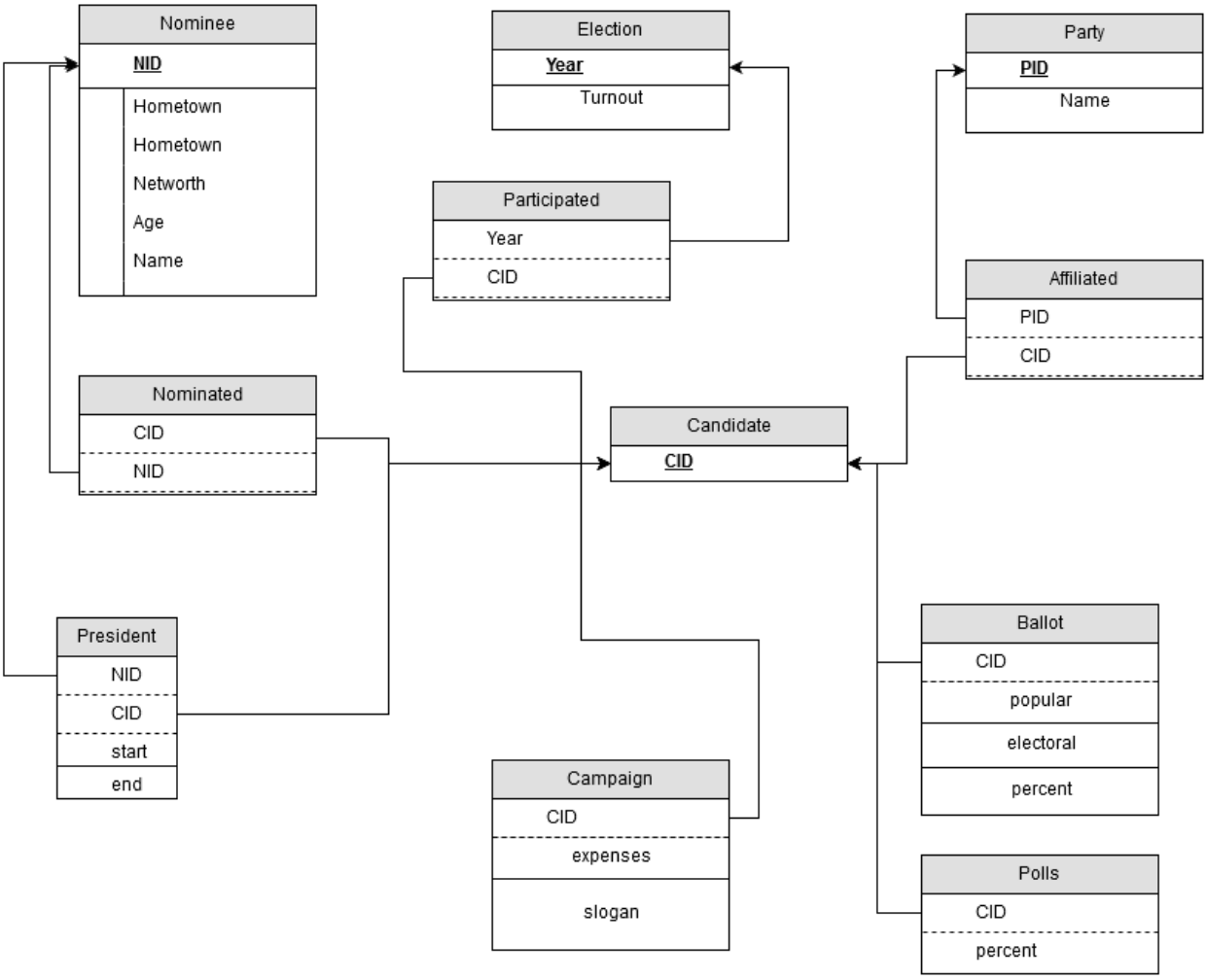

Figure 6: Relational Model (UML)