ccid1: jechmila
ccid2: wwong
ccid3: jtwong
CMPUT291F18

# Mini Project 2 Design Report

## (a) General overview of system and user guide

The system is broken down into three phases -- phase 1, phase 2, phase 3.

**Phase 1**: An input file is parsed and formatted for use in phase 2.

**Phase 2**: The files created in phase 1 are used to create index databases.

**Phase 3**: A query interface allows user to search for ads using the databases create in phase 2.

When the application is started, the user is prompted to enter a filename for the input. This input will be used for phase 1 of this application. If successful, the user is brought to the query interface. There are several commands that the user may enter:

1. **help**: displays a list of available commands in the application
2. **enter query**: allows the user to search with a query
3. **change format**: allows the user to change the information displayed from the query between full ad record and partial ad record.
4. **quit**: terminates the application.

The application will continue to prompt the user for a command until the command quit is called.

## (b) Detailed design of software

The software is again broken down into three phases -- phase 1, phase 2, phase 3.

**Phase 1 (involving fileparser.py and filecreator.py)**: The user enters an input xml file to be parsed and formatted for indexing in phase 2. A separate parser is used to parse the input files. The output files include terms.txt, pdates.txt, prices.txt, ads.txt

**Phase 2 (involving indexing.py)**: The .txt files created in phase 1 are used to create index databases. terms.txt is used to create a B+-tree index te.idx (terms as key and ad id as data). pdates.txt is used to create a B+-tree index da.idx (dates as key and ad id as data). prices.txt is used to create a B+-tree index pr.idx (prices as key and ad id as data). ads.txt is used to create a hash index ad.idx (ad id as key and full ad as data).

**Phase 3 (involving dataretrieval.py)**: A query interface is implemented for the user to enter search queries. Queries are processed through Berkeley DB and the grammar correctness of the user's input is checked with regex. Ads matching the user's input queries are returned to the user.

ccid1: jechmila
ccid2: wwong
ccid3: jtwong
CMPUT291F18

**(c) Test strategy**

Phases were tested individually, making sure each phase works by itself and produces the desired output. To make sure the outputs were correct, we used the example outputs that were posted for our use. After the individual tests, we tested phase 2 with files created from phase 1, and we tested phase 3 with files created from phase 2 to ensure that all our inputs and outputs work together for the desired output. Git was a useful tool for version control in the development of this application.

**(d) Group work and project breakdown**

For the group work breakdown of this project, we divided the different phases amongst ourselves. John developed the data parser used in phase 1. John and Javin worked on phase 1 -- two version were developed but John's version was more functional than Javin's; John's version was decided on. William developed phase 2. All members contributed to phase 3. Github was used to collaborate with each other.