**CMPUT 291 Fall 2019**

**MINI PROJECT II**

**DESIGN REPORT**

**Group Members:**

* + 1. Vanika Dhamija - vanika (1503248)
    2. Jordan Los - los (1550572)
    3. Evan Timms - perretti (1504825)

1. **General Overview of the System with a Small User Guide**

The general overview of the system gives a high-level introduction and may include a diagram showing the flow of data between different components; this can be useful for both users and developers of your application.

The user guide should have instructions for running your code for phases 1, 2 and 3.

Mini Project II of CMPUT 291 aims at teaching the concept of working with data in the physical layer. This is done by building an information retrieval system, using the Berkeley DB library for operating on files and indices. To meet the requirements of this project, programs are written to keep data in files and maintain indices that provide basic searches over data. Another goal of the project is to optimize the efficiency of the programs and queries in terms of running time.

The project is divided into three phases. Phase 1 involves preparing data files for constructing indices in phase 2. This is done for data that consists of a set of emails. Each email record consists of an id, date when the email is sent, a subject, a body, and the fields from, to, cc and bcc. Phase 2 involves building indexes and this is done by sorting the files built in phase 1. The Linux *sort* command and Berkley DB’s hash and B+-tree functionalities are used to do this. Finally, data retrieval is done in phase 3 of the project. A program “main.py” is written to process queries based on the project description.

The program can be run using terminal and typing in following commands for respective phases of the project:

* Phase 1: python3 Phase1.py file.txt
* Phase 2: python3 Phase2.py
* Phase 3: python3 Phase3.py

On running Phase 3, user will be prompted to follow or run commands to perform queries or exit the system.

1. **Description of Algorithm for Efficient Evaluation of Queries**

description of your algorithm for efficiently evaluating queries, in particular evaluating queries with multiple conditions and wild cards and range searches and an analysis of the efficiency of your algorithm

**Phase 1: Preparing Data Files**

DESCRIBE FUNCTIONS

**Phase 2: Building Indexes**

Sorted files, namely, terms.txt, emails.txt, dates.txt and recs.txt, are obtained as a result of running phase 1 files. These files create the following four indices:

1. A hash index on recs.txt with row ids as keys and the full email record as data
2. A B+-tree index on terms.txt with terms as keys and row ids as data
3. A B+-tree index on emails.txt with emails as keys row ids as data
4. A B+-tree index on dates.txt with dates as keys and row ids as data

DESCRIBE FUNCTIONS

**Phase 3: Data Retrieval**

This phase of the project allows the user to open the databases and process queries. The default output format of each query is the row id and the subject field of all matching emails. The user can change the output format to full record by typing *"output=full"* and back to id and subject field only using *"output=brief"*.

DESCRIBE FUNCTIONS

1. **Testing Strategy**

The testing strategy discusses your general strategy for testing, with the scenarios being tested and the coverage of your test cases.

Started building the program by having all the set-up data in place and writing the code for one phase at a time

* Started building the program by having all the set-up data in place and writing the code for one phase at a time and testing it subsequently. This allowed to build a bug-free program in small steps.
* All the code files from the three phases are later combined to implement an executable program which is tested using the given data files.
* It was ensured that the project code works well with the python version on the lab machine by testing it on the ab machine.
* OTHER STRATEGIES USED (“Try and Except” block, print statement to debug a program or Write test cases to test code)

1. **Group Work Break-Down Strategy**

The group work strategy must list the break-down of the work items among partners, both the time spent (an estimate) and the progress made by each partner, and your method of coordination to keep the project on track.

Break down of work items:

* Evan:
* Jordan:
* Vanika:

Estimate of time spent and progress made by each partner:

* Evan:
* Jordan:
* Vanika:

Method of coordination to keep the project on track:

* The group met in the beginning of the project to work together on Phase 1 and 2. The overall strategy of the project was also discussed and work for Phase 3 was divided among the group members.
* The group also met on the submission day to go through the code together, refactor it, rename files for better readability and therefore, to finalize the submission.
* The code was pushed on GitHub so the members could see the current state of the project at all times and review each other’s work as well.
* All the other communication was done in a group chat, including modifying work distribution as necessary and tracking progress.
* The design document was prepared by Vanika and was shared as a google doc for further collaboration.

1. **Assumptions made in the Project**

Include any assumption you have made or any possible limitations your code may have.

(No further assumptions were made in this assignment ???)