Names: Kevin Gutierrez, Kiran Kadariya, Sagar N, Abdul

CSCI 331-54

Professor: Andrew Anda

Date: 4/8/25

Zip Code Group Project 2.0 Design Document

1. The project 2.0 codebase will be used as a basis for project 3.0. Using 2.0 as a basis, a blocked sequence set file will be defined by parsing the data file from 2.0, such that each block of the sequence set will be of equal size.
2. The sequence set file requires some config – this will be supplied using command line args.
3. The sequence set in (1) will be processed using buffer classes from 2.0
4. Two new classes will be created - BlockBuffer and RecordBuffer  
   Notice! The following will represent the data structures – I wrote it in C# as that’s what I find to be read the easiest in place of pseudocode; of course the related data structures in c++ can be deduced from the following relational data units (classes):

A screenshot of a computer program

AI-generated content may be incorrect.

1. Using the blocked sequence set file from (1), we’ll use the block sequenced file from (1) to create a data file like in project 1.
2. Then we’ll create a dump method that will aggregate the zip codes, such that the dump will look like a linked list, except instead of hex memory addresses, it’ll use a relative block number (essentially an incrementing primary key integer index – **for blocks**). The contents between each key will be the key for each **record,** and since each record contains k fields but exactly one key, there is a one to one relationship between keys and records!
3. A simple index file will be created containing ordered pairs of keys – the highest of each block and block numbers (RBN’s)
4. Using the simple index, we’ll create a readable dump.
5. Simple primary keys will be used to display zip code data from all zip codes listed in the command line, in the structure of a key of highest key in block and a value of relative block number. We will add functionality to search based on these files.

Data structures:  
