# Preliminary Design Document: Blocked Sequence Set Generation Program

## Objectives:

**CSV to File Structure Conversion:**

* + Convert an XLSX file to a CSV file.
  + Transform the CSV file into a file structure format with length-indicated records and comma-separated fields.
* **Blocked Sequence Set Generation:**
  + Generate a blocked sequence set file from the data file created in Group Project 2.0.
  + Implement command line options for the header file and blocked sequence set data file.
* **Buffered Processing:**
  + Process the blocked sequence set file sequentially using buffer classes.
  + Use block buffer and record buffer classes to read and unpack Zip Code Records into a sorted container.
* **Dump Methods:**
  + Create two blocked sequence set dump methods, one listing blocks by physical ordering and the other by logical ordering.
  + Display the aggregated Zip Codes with predecessor and successor RBN links.
* **Indexing:**
  + Create a simple index file containing ordered pairs of keys and block numbers.
  + Generate a readable dump of the simple index.
* **Search Program:**
  + Develop a blocked sequence set search program with command line options for the data file and Zip Code record to search for.
  + Display a message if the Zip Code record is not in the file.
* **Test Run Demonstration:**
  + Create and run a search test program for valid and invalid Zip Codes.
  + Implement a record addition and deletion test program, logging events such as block splits and index modifications.
* **Documentation:**
  + Document C++ source code extensively with comments and Doxygen tags.
  + Generate a Doxygen PDF of class and application program code.
  + Create a user guide explaining program usage, command line options, and expected output.

## Requirements:

### Header Record Architecture:

* File structure type: Blocked sequence set with comma-separated fields and length-indicated records.
* Version of the file structure type.
* Header record size.
* Number of bytes for each record size integer.
* Size format type (ASCII or binary).
* Block size (default to 512 Bytes per block).
* Minimum block capacity (default to 50%, except for the last block).
* Index file name and schema information.
* Record count and block count.
* Count of fields per record.
* For each field: name/ID, type schema, format to read or write.
* Primary key field indicator.
* RBN link to the block avail-list and active sequence set list.
* Stale flag.

### Block Architecture:

#### Active Block:

* Count of records (> 0).
* Links to preceding and succeeding active blocks.
* Set of records ordered by key.

#### Avail List Block:

* Count of records (== 0).
* Link to succeeding avail block.
* All other bytes overwritten with blanks.