

## Programming Assignment 2

**Due Date:** May 13, 2022, 23:59

The task of this programming assignment is to extend your C++ implementation from Programming Assignment 1 for a **registration, queueing and reporting system** for mass medical treatment (such as vaccination). The extensions concern the efficient implementation of index structures for relations that are represented by lists of blocks, to implement queries on these relations in an efficient way and to produce graphs as output.

EXERCISE 1. Define and implement the basic data structure for the storage of relations:

- (i) Design a **relational database** schema with relations PERSON, MEDICAL\_STATUS, REGISTRATION and TREATMENT capturing all data that are to be collected during registration, scheduling and treatment processing.
- (ii) In addition, assume that there are **three** types of treatments a person may register for, **each with different priority rules**.
- (iii) Represent each of these relations in a data structure that uses an **ordered sequence of blocks**, where each block is realised by an array.
- (iv) Permit an **additional overflow block** for each block in the sequence.
- (v) Implement **operations**
  - for the insertion and deletion of records;
  - for the reorganisation of blocks by merging, sorting and splitting;
  - for the retrieval of a record from a relation.

EXERCISE 2. Define and implement the basic operations on the data structure from Exercise 1:

- (i) **Modify the operations** for registration, priority calculation, appointment scheduling and reporting using the database schema.
- (ii) Create **file output** for the various reports.

In a third step you are to realise index structures that will support the fast access to the relations to facilitate the processing of the queries. For this you have to **identify primary and secondary keys and implement B+-trees**, B-trees or hash tables.

EXERCISE 3. Implement index structures on the relations above and integrate them into the query processing:



- (i) Identify for each of the relations above a **primary key** and organise the data structure in Exercise 1 accordingly.
- (ii) Implement index structures for your primary keys **using B+-trees**.

- (iii) Realise your operations from Exercise 1 using your **index structure**.
- (iv) Identify at least **one useful secondary keys** and implement an index structure for it using a B-tree.
- (v) Integrate your index structures into the processing of the queries **in Exercise 2**.

**total points: 100**