

CSC3170

Project Abstract

Genshin Team

120090590 Chen Feiyang

120010042 Tao Zhen

120090460 Zhu Luokai

120090582 Chen Yiwen

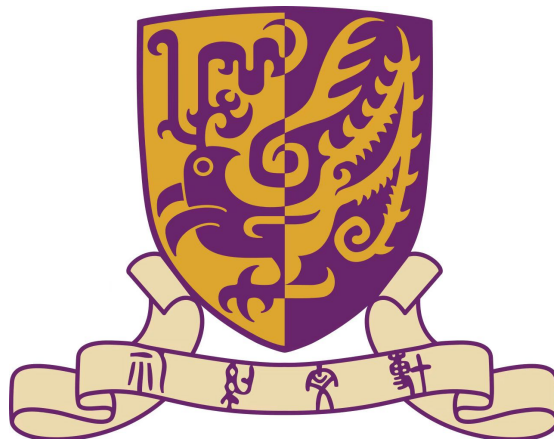
119010166 Li Zhuoyi

120090447 Wang Yiwen

120090089 Wang Yuqi

120090588 Xiao Weizhao

2022. 12. 5



Contents

1	Project Background	3
2	Project Overview	4
2.1	Basic Contents	4
2.2	Basic Functions	5
3	Project Extension	6

1 Project Background

In this project, we will attempt to create a database management system (DBMS) using static programming language (Java). We will give the database the basic functions required, while adding more convenient and efficient instructions based on our technology and ideas.

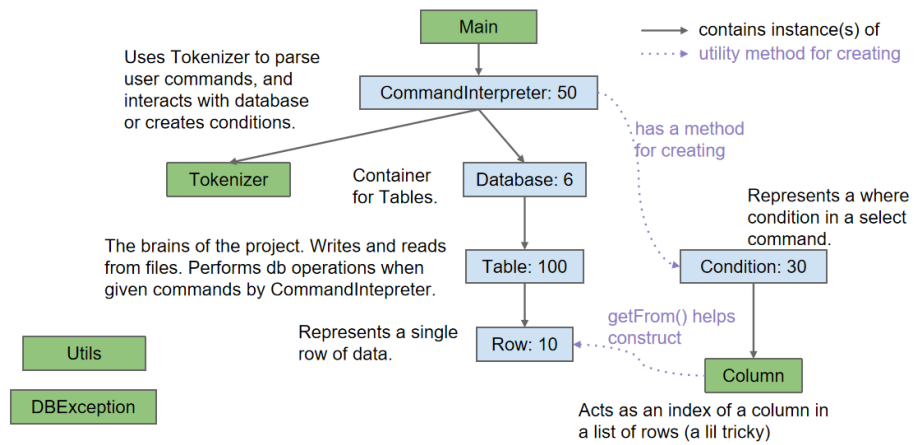


Figure 1: Project Organization

2 Project Overview

2.1 Basic Contents

Our project object is a miniature relational database management system (DBMS) that stores tables of data in which we will set some number of labeled columns of data information. Our system will include a very simple query language for extracting information from these tables.

students

SID	Lastname	Firstname	SemEnter	YearEnter	Major
101	Knowles	Jason	F	2003	EECS
102	Chan	Valerie	S	2003	Math
103	Xavier	Jonathan	S	2004	LSUnd

Figure 2: Sample Table1, students

schedule

CCN	Num	Dept	Time	Room	Sem	Year
21228	61A	EECS	2-3MWF	1 Pimentel	F	2003
21231	61A	EECS	1-2MWF	1 Pimentel	S	2004
21229	61B	EECS	11-12MWF	155 Dwinelle	F	2003
21232	61B	EECS	1-2MWF	2050 VLSB	S	2004
21103	54	Math	1-2MWF	2050 VLSB	F	2003

Figure 3: Sample Table2, schedule

enrolled

SID	CCN	Grade
101	21228	B
101	21105	B+
101	21232	A-
101	21001	B

Figure 4: Sample Table3, enrolled

2.2 Basic Functions

We will prepare the following basic functions under the framework of the database system built in the project.

- **Database** The function first creates an empty database. Then we can use the get function to enter the table name to quickly find the return table or null, or use the put function to legally change the table name.
- **Row** We will create a class named row. Users can set and update the value of the row through the function, and return the relevant information of the row (e.g.number,value). We also added equals function to judge by hashcode.
- **Table** This function contains the basic functions associated with Table. Users can find and return labels, values, and so on for specified rows and columns. We set up the function of reading and output table, and print the search results in standard form; At the same time, the filter table function is set to filter the tables that meet the requirements based on the row and column values.
- **Interpreter** We create an interpreter that uses the recursive descent principle, converts the bnf syntax, unties the input side, and uses tokens to create statements from the remaining sequences and perform operations, returning values. Specifically, we will prepare table(eg.create, exit, insert, load, store, print, select) for the statement.
- **Condition** We will set the corresponding relation for different conditions (e.g.val2,col2). We also added a test program to judge if rows satisfies all conditions.

3 Project Extension

We would like to add some extensions of our own ideas to the features required by the project.

- **Basic mathematical functions** We will add "Max", "Min", "Avg" and other basic mathematical functions to calculate the statistical information of rows and columns
- **More information** We will set more signature information(e.g.primary key) for the table, make it easier for the user to find. We will also try to add sort to rows and columns. In addition, we will also try to incorporate group statistics features such as group by.
- **Multi-select** In order to get the table with the required information step by step, we will expand on the basis of select so that the value returned by the lookup can be multi-tiered and step-by-step.
- **Join of tables** We will use different connection modes to connect the required information according to user requirements (e.g.left join, right join).
- **GUI** We will try to create a fancy and user-friendly GUI that makes it easy for users to select the features they need, with helpful tips (e.g. Show possible paths as users type).