



# Documentation for advanced use of the Database -Gentlema(n)

# Acknowledgments/About This Documentation:

This Documentation is provided by Gentlema(n) for a quick introduction to the Database I have.

Note: This Database is and was never intended to be the best.

I just was having fun creating this and hopefully you do the same.

---

## Table of Contents

Preface	i
About This Documentation .....	i
<a href="#">UML</a> .....	3
<a href="#">How does the database and CLI work</a> .....	6

# UML

Note: I'm not fully experienced with UML, so I might have mistaken somewhere, which I would love to have a review about 😊

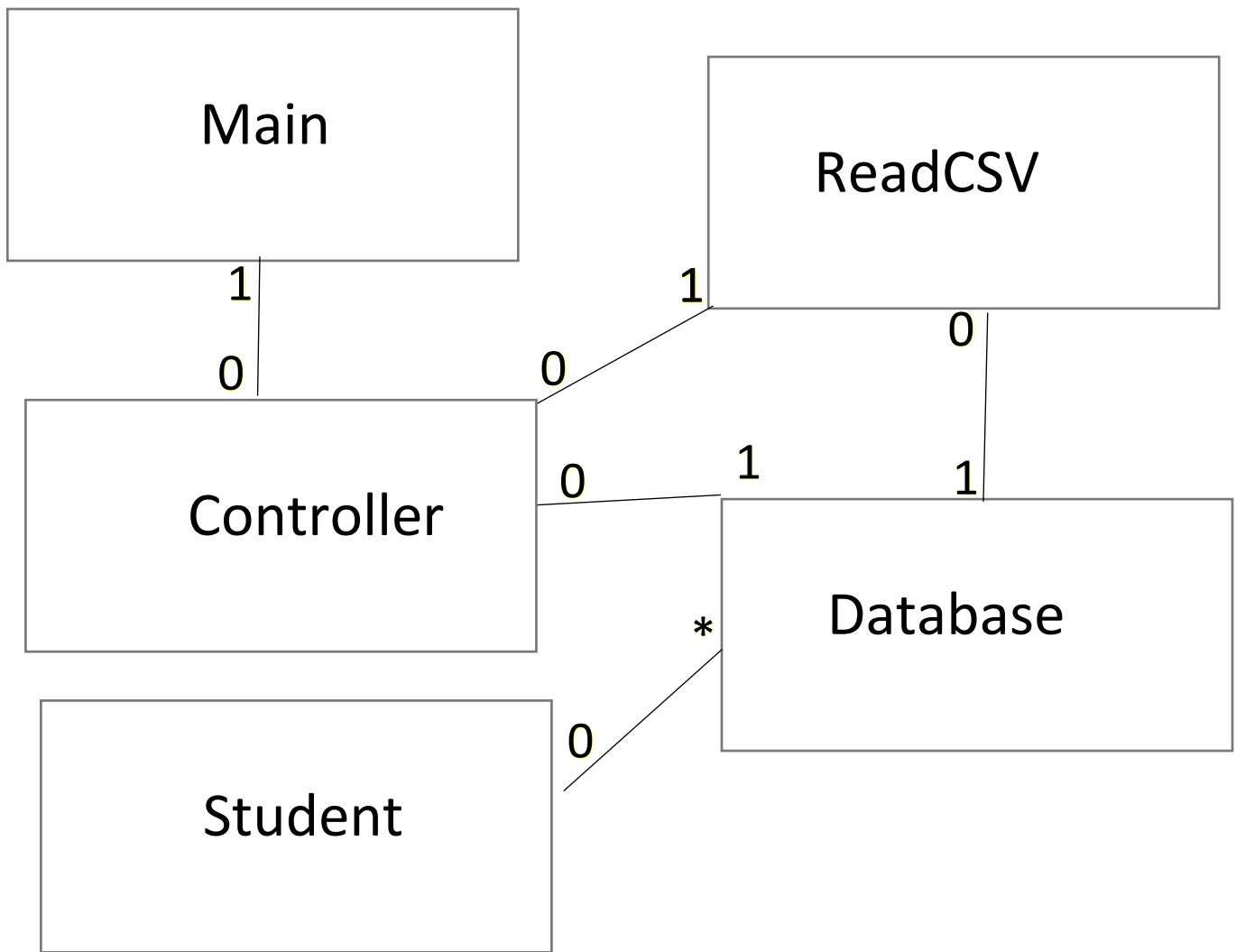
Database
<div>-Students:Student</div> <div>- backup:Student</div> <div>-fivedigits:int</div> <div>-year:int</div>
<div>+int: matgenerator(numberOfSemester : int)</div> <div>+void: adduser (lastname: String,firstName: String, numberOfSemester: int)</div> <div>+ void: adduser (lastname: String,firstName: String)</div> <div>+Boolean: exists(matknr: int)</div> <div>+void: delete(matknr: int)</div> <div>+Student: isdeleted(matknr: int)</div> <div>+void: retrieve(matknr: int)</div> <div>+void: save(fileLocation: String)</div> <div>+void: saveDeleted(fileLocation: String)</div> <div>+void: print()</div> <div>+void: printDeleted()</div> <div>+void: printOne(matknr: int)</div> <div>+void: update(lastname: String,firstName: String,matknr: int, numberOfSemester: int)</div> <div>+void: update(lastname: String,firstName: String, matknr: int)</div> <div>+void: printcommands()</div> <div>+void: command()</div> <div>+void: command(csvInput: String)</div>

Student
-name:String - firstname:String -matknr:String -numberofSemesters:int
//Getters and Setters should be included here! +Student(name: String,firstName: String, matknr: int, numberOfSemester : int) +Student(name: String,firstName: String, matknr: int)

ReadCSV
-csvInput:String
+void : readFile(fileLocation: String,commandNumber: int)

Main
+void : main(String[] arg)

Controller
+void : controller()



# How does the database and CLI work

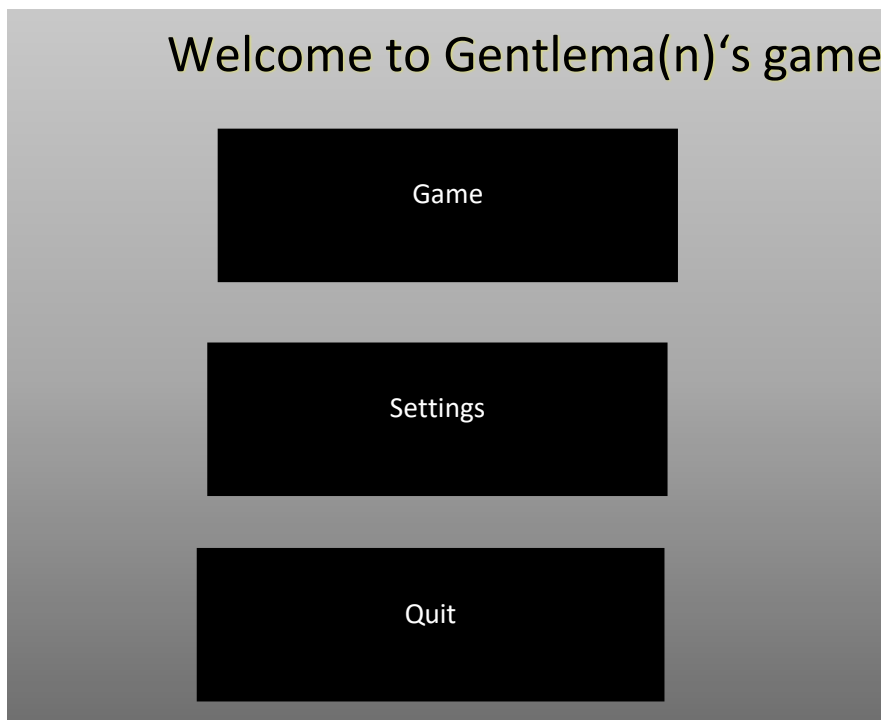
This database was made to be easy to use.

Now we start with the main function, which is in the “Main” class.

The main function is the “main menu”.

For example when you open a game, the game will automatically load you into the main menu, which might be a picture and some clickable text.

Something like this:



Now since you understand the point of the “main menu” let me show you how my CLI looks like

```
Main Menu:
Do you have an existing CSV file or do you want to create a new database?
Enter no/na to create a new database and yes/y to read an existing CSV file
To quit enter ('quit'):
>
```

A very simple and useful one.

“Yes” will load you to the file -> ReadCSV.java, which you can use if you have a .csv database to use

But before you go to the database you need to specify what kind of “improvements” you need.

For example you might have already a database and you want to modify yours by letting this App to read a simple .csv file and it you can save the hard “typing warrior” you have inside of you.

```
>yes
Type a number please
(1)Do you want to addusers to the database through a file?
(2)Update existing users through a file?
(3)Delete existing users through a file?
(4)Retrieve existing users through a file?
>
```

Now after typing “yes” you will have different options to choose.

You can addusers, retrieve, update and delete just using one button and your .csv file

So let’s continue with the app and type 1 (Since I don’t have a database I will load one)

```
Where is your file located at (global)? (the database should be a csv file):
>D:/CSV/create.csv|
```

Now I will just type the file location. I would suggest you to type the global one.

On Linux it will look like something like this : /home/CSV/create.csv

After typing your file location, the app will load you to the Database.java.

You will be welcomed with this pop up:

```
>Entering CMD...
CMD:
This are the following commands to use:
To create a user use the command: ('create' lastname;firstname:numberOfSemesters)
To retrieve a user use the command: ('retrieve';matriculation number')
To update a user use the command: ('update';name;firstname:numberOfSemesters)
To delete a user use the command: ('delete';matriculation number)
To print all users use the command ('print-all')
To print a user use the command ('print' matknr)
To save the file use the command ('save') Note:the database should be a csv file:
To save the deleted-database use the command ('save-deleted') Note:the database should be a csv file:
To see more type ('info')
To quit use the command: ('quit')
>
```

Now as you can see you have a lot of commands you can use to improve the user’s experience, but for now I will “print-all” to show you how they look like

```
>print-all
id: 0 Lastname: NoWay, Firstname: ThisIsEasyyy, Matriculation number: 22200001, number of semesters:2
id: 1 Lastname: Testing, Firstname: Yikes, Matriculation number: 22200002, number of semesters:2
id: 2 Lastname: HowEasy, Firstname: man, Matriculation number: 22200003, number of semesters:2
id: 3 Lastname: Weird, Firstname: dex, Matriculation number: 22100004, number of semesters:1
id: 4 Lastname: Database, Firstname: BeLookin, Matriculation number: 22200005, number of semesters:2
id: 5 Lastname: HowIsThis, Firstname: Working, Matriculation number: 22100006, number of semesters:1
```

Here you can see all users I have created using only a couple of inputs.

The [create](#).csv file I had looks like this:

NoWay	ThisIsEasyyy	22200001	2
Testing	Yikes	22200002	2
HowEasy	man	22200003	2
Weird	dex	22100004	1
Database	BeLookin	22100005	2
HowIsThis	Working	22222222	1

As you can see I have first of all the “firstname” and then the “lastname” the matriculation number and then the number of semesters.

Now you should know that the matriculation number is [auto-generated](#) and does not effect the database at all.

So let’s “quit” from the CLI go to the [“main menu”](#) we have and try to update the database

```
>quit
Quitting... to main menu
Main Menu:
Do you have an existing CSV file or do you want to create a new database?
Enter no/na to create a new database and yes/y to read an existing CSV file
To quit enter ('quit'):
>y
Type a number please
(1)Do you want to addusers to the database through a file?
(2)Update existing users through a file?
(3>Delete existing usesrs through a file?
(4)Retrieve existing users through a file?
>2
Where is your file located at (global)? (the database should be a csv file):
>D:/CSV/update.csv
```

As always we get redirected to the CLI and we “print-all” and we have now

```
>print-all
id: 0 Lastname: updated, Firstname: updated, Matriculation number: 22200001, number of semesters:1
id: 1 Lastname: updated, Firstname: updated, Matriculation number: 22200002, number of semesters:1
id: 2 Lastname: updated, Firstname: updated, Matriculation number: 22200003, number of semesters:1
id: 3 Lastname: updated, Firstname: updated, Matriculation number: 22100004, number of semesters:1
id: 4 Lastname: updated, Firstname: updated, Matriculation number: 22200005, number of semesters:1
id: 5 Lastname: updated, Firstname: updated, Matriculation number: 22100006, number of semesters:1
>
```

To understand what happened let me show you the update.csv file

updated	updated	22200001	1
updated	updated	22200002	1
updated	updated	22200003	1
updated	updated	22100004	1
updated	updated	22200005	1
updated	updated	22100006	1

If you look clearly you can see that I changed the first/last-name to updated and the numbers of semester too, but I can’t change the matriculation number by any chance not at all!



You can delete and retrieve users using a csv file, but I won't show them due to your valuable time.

They will be included in the files too, so you can have a good and clear look at them to see how astonishing Java can be.

Now if we type [no](#) instead of [yes](#) at the very beginning in the "[main menu](#)" or even type yes and select file you will be redirected to the [Database.java](#)

In CLI you might have seen the [command](#) that are available.

Will abbreviate

lastname = ln

firstname = fn

numberofsemesters: nos

matriculation number: mn

Command	Parameter	Description
Create	ln;fn;nos	Creates a user
Retrieve	mn	If a user is deleted, you can retrieve him
Update	ln;fn;mn;nos	Update user's name,firstname and number of semesters, but mn
Delete	matriculation number	
Print-all	None	Print all users
Save	None	After typing "save" and pressing "enter" you have to specify the folder and then the csv file-name. And all of the users will be saved there.
Save-deleted	None	After typing "save" and pressing "enter" you have to specify the folder and then the csv file-name. And all of the deleted users will be saved there.
info	None	Prints all available users
quit	None	Quits to <a href="#">main menu</a>

## Links

<https://unsplash.com/s/photos/database>