Budapest University of Technology and Economics

WEB-based Laboratory

Written report

Beerculator

Authors:
Alvaro Gonzalez,
Maud Leray,
Patricio Sanchez,
Tomás Sekanina

Supervisor:
Dr. János Hamar

Contents

1 Introduction

As students, we often hang out with our friends in pubs or in nightclubs. In those places, we usually have a few drinks, most of them containing alcohol. We are aware that this can be risky, and we also know that we should not exceed a certain amount, especially if we need to drive after the party. But the problem is that most of us do not know their limits, and do not realize how long it takes for their body to remove the alcohol.

To create a more alcohol-conscious society, there should be tools to inform the general crowd about the quantity of alcohol a person can ingest and the effects it will have on him or her. This is why we created Beerculator, an interactive tool which calculates the blood alcohol concentration based on the drinks that a person has had in a certain amount of time. It will also indicate how long the person has to wait before being sober again.

2 Requirement analysis

Bob (made-up client for the project) consumes alcohol and can feel its effects in his bloodstream, but does not know when to stop or if he is on a safe-to-drink amount of alcohol. Bob requires a simple interactive tool that he can use on the go to check his alcohol level without being too specific about the details.

Bob should not be the only one to access this tool, so it should be available on the Web. He does not need the tool to store informations about previous alcohol level calculations, he just wants it to dynamically calculate his current alcohol level. He also wants to know how long it is going to take for his body to remove all this alcohol, for example because he absolutely needs to drive tomorrow morning to go to work. So when he uses Beerculator, Bob would like to be able to perform all the actions described in FIGURE 1.

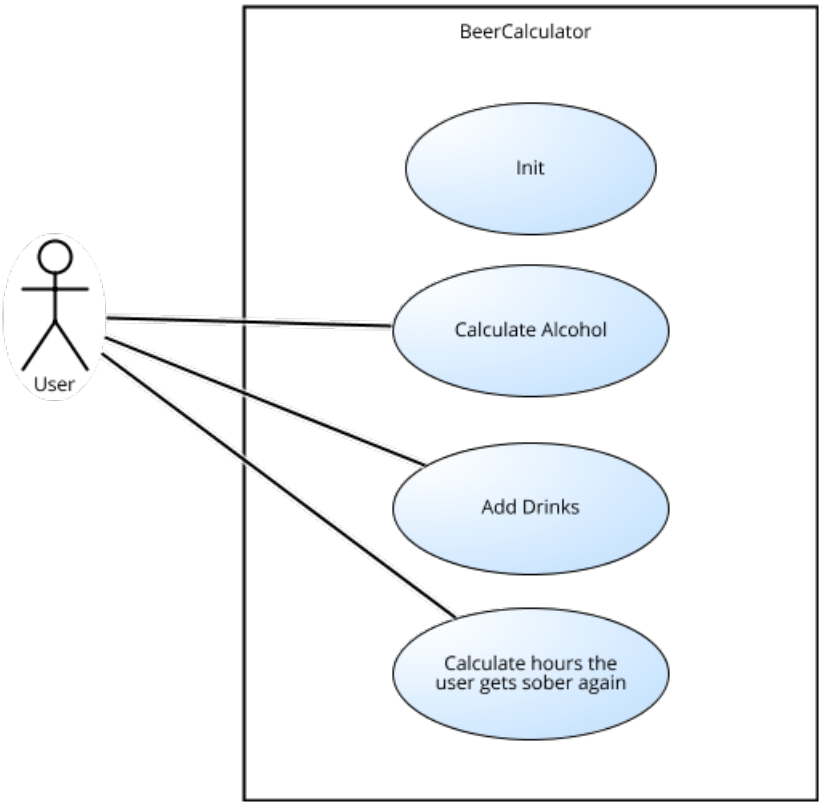

Figure 1: User case diagram for Beerculator

In order to perform the calculations, Beerculator needs to know a few informations about Bob: his gender, his weight, the amount of alcohol he has consumed, and also the time when the drinking started. This will be detailed in SECTION 3.

3 Specification

3.1 Needed informations

In order to calculate the user's blood alcohol concentration (BAC), we need a few informations about him and about his drinking. More precisely, we need the elements listed below:

- Gender (M/F);
- Weight;
- Number of hours since the drinking began;
- Amount of alcohol.

3.2 BAC calculation formula

To calculate the blood alcohol concentration, we will use the Widmark formula, which is the following:

$$BAC = A \div (R \times M) - 0.015 \times H \tag{1}$$

In this formula, we have:

- BAC : Blood Alcohol Concentration;
- A : Alcohol ingested, in grams;
- R : Ratio (0.70 for men, 0.55 for women);
- M : Body weight, in kilograms;
- H : Number of hours since the drinking began.

To obtain the value of A, we use this formula:

$$A = (V \times P \times 0.8) \div 100 \tag{2}$$

In this formula, we have:

- V : Volume of alcohol in milliliters;
- P : Alcohol degree, in percentage.

3.3 Elimination of alcohol

The median rate of decrease in BAC is considered to be 15 milligrams per cent (mg%) per hour. In our calculations, we used the approximations listed in TABLE 1.

BAC value	**Hours until sober**
0.016	1
0.05	3.75
0.08	5
0.10	6.25
0.16	10
0.20	12.5
0.24	15

Table 1: Hours until sober depending on the BAC value

4 Solvable tasks and team management

After a few meetings to determine the requirements and the objectives of the project, the implementation of Beerculator has been divided into several groups of solvable tasks, as detailed in TABLE 2.

Group	**Tasks**
Diagrams	UML for Java UML for databases User interaction
Databases	Create tables Set up methods Define relations Write script
Server	Define Java server Create Java beans
Calculation	Find formulas Implement them
Design	User interface

Table 2: Solvable tasks for the implementation of Beerculator

Some other tasks have been completed in parallel for the project, but are not referenced in TABLE 2 because they are not directly linked to the development of Beerculator. For example, writing the final report was also part of this web-based project, and every member of the team contributed to it.

We often worked all together and we had very regular meetings, so everyone was always aware of the state of the project and of the remaining work. And of course if anyone had a problem on its task, we helped each other so we all contributed to the whole project. But in general, each group of tasks in TABLE 2 had a « manager » and the distribution was the following:

- **Alvaro**: Diagrams;
- **Maud**: Calculation;
- **Patricio**: Server;
- **Tomás**: Databases.

The last group, Design, is not in the previous list because it had no real responsible, everybody in the team equally contributed to it.

5 Selected technologies and alternatives

5.1 Selected technologies

The two main technologies used in this project are Java and XHTML. Java was used to code the main program (for testing) but most importantly to perform all the calculations. XHTML was used to define the web page, which user interface was designed with CSS3, Javascript, and the bootstrap library.

PostgreSQL and its JDBC (Java Database Connectivity) driver were used to create, store and manage the various databases of the project. This has been recommanded at the beginning of the semester.

We also needed Glassfish4 to create and run the application server, and to synchronize it with the project.

We chose to work with the JSF (JavaServer Faces) framework, associated to Facelets, to develop the Web components.

5.2 Alternatives

The selected technologies introduced in SUBSECTION 5.1 are not the only ones that we could have used. This subsection is here to present a few alternatives.

There are many options to handle databases, for example we could have used MySQL instead of PostgreSQL.

To build the user interface, it would have been possible to use PHP with CSS and HTML.

Finally, we could also have used JSP (JavaServer Pages), and not JSF, to implement the web page.

6 Design

6.1 Program structure

6.1.1 Databases

There are three databases in the project, which are the following ones:

- drinks: contains all the predefined drinks that the user can select;
- drink_records: contains the drinks selected by the current user;
- users: contains all informations about the different users.

We also needed queries to fill the « drinks » database, they are all written in a SQL script called list_drinks.sql.

6.1.2 Java

The project consists in three main classes, Drink.java, DrinkRecord.java and User.java. Their respective UML representations are given by FIGURES 2, 3 and 4.

The calculation methods, calculateBAC() and hoursUntilSober(double), are implemented in User.java. They are based on the formulas previously introduced in SECTION 3.

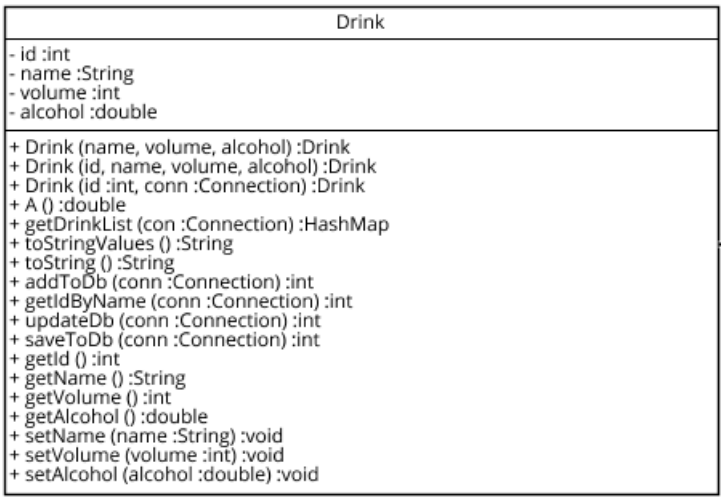

Figure 2: Class Drink.java, its attributes and its methods

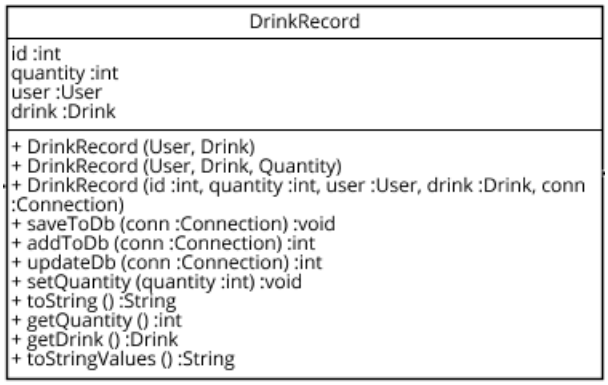

Figure 3: Class DrinkRecord.java, its attributes and its methods

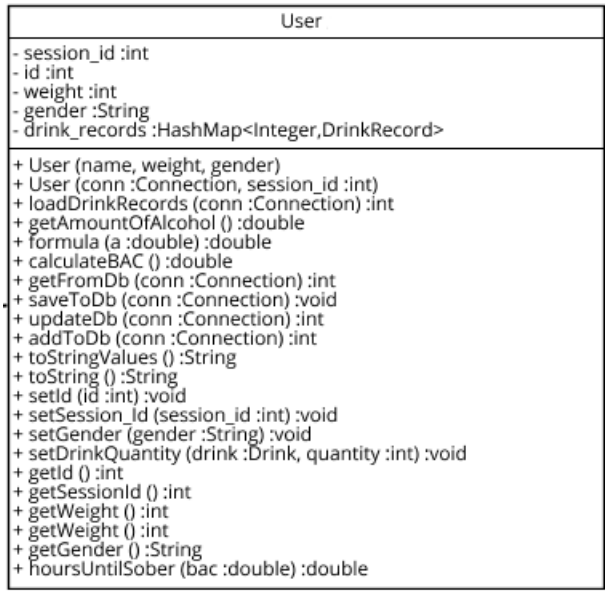

Figure 4: Class User.java, its attributes and its methods

6.1.3 Web content

The web content is located in the file Index.xhtml. It is used to generate different URLs based on the session ID. This means that instead of creating a new page for each user, it is just redirected to the page with the right session ID.

The web content also includes a folder called WEB-INF, that contains two files: web.xml and faces-config.xml. They are used to manage the server and to handle the Java beans.

6.2 User interface

Three main elements were needed in Beerculator's interface : one to get the user information (weight, gender, etc), one to display the list of drinks and one to show the results. The user

interface can be seen on FIGURE 5.

Figure 5: User interface of Beerculator

On the top left part of the interface, the user can add all informations that Beerculator needs from him. He can indicate its gender (male or female) via two check boxes, and its weight thanks to a text box. Another text box is available for him to tell the time that he started drinking. He can also save all those informations, in case he has to close the page and wants to come back later.

On the right side of the web page, the user can see a list of predefined drinks (with their volume and their alcohol percentage). With each alcohol there are two squares: one green and on red. The user can click on the green one to add one item of the corresponding alcohol to his list of drinks. If he clicks on the red button, one item of this alcohol is removed from the list of drinks. He can also save his list of drinks, for instance if the party is not over and he intends to add some more later in the night.

When he is done, the user can click on the « Calculate » button so that Beerculator starts the computation. The results appear on the bottom left corner, in the « Calculation » tab. First the BAC value is indicated, and below it the number of hours before the user gets sober again.

7 Testing instructions

When you get to Beerculator's web page, the first thing to do is to enter your personal informations. To do so, in the « User data » tab, you have to indicate your weight in kilogramms and then your gender (Male/Female). You should also indicate what time you started drinking (for example, 8pm). If one of these is missing, Beerculator will not be able to perform the calculation. Once you are done here, you should click on the « Save » button.

After that, you can move on to the right side of the screen and start adding drinks to your list. To do so, you have to click on the green squares corresponding to the drinks you have had. If you made a mistake, for instance if you added the wrong drink, you can always click on the associated red square in order to cancel and remove this drink. Then, to be sure not to loose your list, you should click on the « Save » button.

Finally, you can launch the computation thanks to the « Calculation » tab. The BAC value is indicated in the first text block, and it is followed by the time (in hours) that you have to wait before being sober again.

8 Conclusion

9 Further plans

Beerculator is already a very useful tool, but it could still be extended with some new functionalities.

New drinks Beerculator allows the user to choose what he has consumed among a list of predefined drinks. It would be nice for him to be able to customize the application, and more precisely it would be interesting if he could add some new drinks to the database, for example his favourite ones.

Mobile version Beerculator has been coded for a desktop use, it could be interesting to make its interface responsive so that people can use it on their mobile phones. It could also become an Android application that users can download on their device and then customize.