**Teona Ristova**

**Vladyslav Cherevkov**

**Mikolaj Debicki**

**Test Report**

The document below outlines the major problems (bugs) we encountered in our development process and what steps we took to resolve them into stable features. All of the major issues that posed a difficulty were either on the SQL database side, within the communication channel between our code and the database, or because of not properly restricting input that were to be later on used for calculations.

**Problem:** There was a problem with the connection between our database and the python code. The connection used to sometimes work and other times it would randomly disconnect. This issue especially became a problem when trying to register a new user with the same email as the one used beforehand to register another user (test case 001). In that case the app would consistently crash.

**Solution:** Previously, the process of establishing this connection between our code and the database used to be done (stored) in a global variable (cursor). The way we fixed it was to make a function that returns the connection as a local variable. Meaning that now when we want to establish a connection we just need to call that function that works in a local scope.

**Problem:** we encountered a problem when trying to calculate the user’s BMI in the case that user accidentally (or on purpose) decided to provide invalid input in the form of text instead of a number for weight and height (necessary parameters). In that scenario, when the user entered a string, we would use it right away for calculation purposes and the application would crash.

**Solution:** We were able to eliminate this problem by checking if the input is a number using python’s .isdigit() and .isfloat() functionalities. Now if the user enters something that is not a number, the app will not try to calculate the BMI (therefore not crashing) but instead prompt the user to enter valid information.

**Problem:** As discussed in the test case 006, we also encountered some issues when working with the Exercise table in our SQL database. The app would crash because of unsuccessful communication with the database in the form of trying to insert a duplicate primary key. This happened due to the fact that we initially used the date + login as the primary key for this table. However, the problem with that arose when the same user tried to insert their daily information twice in the same day.

**Solution:** We restructured the Exercise table to have a unique primary key that does not depend on the date of entry which fixed this issue and has not produced any problems since.