**Bauhaus-Universität Weimar**

Digital Engineering Program

# SOFTWARE ENGINEERING

Project Handover

**PASSWORD GENERATOR**

**Group W**

Group Members:

* Lakshmi Narasimha Jinkala (127091)
* Samip Timilsina (127193)
* Nisarg Maurya (123414)

Implementation:

We have received the password generator from Group F to test the code they have used for this application (Password Generator). We have done black box testing for only login system because it is the only part in the code where we can expect output with string datatype. We can use black box testing for the methods used in the AppGUI.java file after the user login to the application. But mostly the test cases will be Boolean and when we checked the whole code it is mostly working fine. Black-box testing focuses on evaluating the system’s behaviour without delving into its internal implementation details. We operate through the system’s interface (inputs and outputs) to verify functional and non-functional requirements and identifies interface-level bugs and ensures that the system meets specified requirements. Whereas white-box testing deeply scrutinizes the system by examining its internal code, logic, and structure. For white-box testing we have changed some code for the user authentication in the code. (the suggestion for the next team will be try to add database for User Authentication).

For white-box testing, we have created various JUnit test classes that test functionality of various components within the project. While checking the code we have seen that all the buttons don’t have that much affect in the code and the main test cases has to be checked for passwordEngine.java file so we did the same. We have implemented test for password length(testPasswordLength) where it will check if the password is generating as per the input value of integer value, Lowercase(testIncludeLowerCase) where it will check if the password contains lower case letters, Uppercaser(testIncludeUpperCase) where it will check if the password contains Upper case letters, action performed in the infoGUI.java file (testActionPerformedReturnButton) where after logging in whether the loginpage is hiding or not, colors(testInitialColors) where the method will check for the change in the colors based on the selection of the colors test for Event Creation and Modification, test for checking of numbers in the created password, test for checking special characters in the password generated.

For Black-box testing, we have checked the functionality of the code by selecting various elements in the window of loginpage, appGUI page, passwords storage page and has noted the test cases briefly in the test-description document, it is as follows, test for User-Registration where for register the input is taken and it is pass, for blank user registration it has to show invalid username, which was not showing as expected so it is failure, test for Login – for the user to login he has to be previously registered for this test Pass for strings and numbers, Failure for blank user login, test for Generating passwords, checking the check boxes for generating password, checking the functionality of buttons in the appGUI window (buttons like Copy to Clipboard, Storage of Password, Languages).

Suggestion for the next team:

Try adding:

1. User authentication with the database,
2. In the AppGUI.java file which is the next page after the login page, try adding logout button to go back to the main login page,
3. More colours to the page,
4. In the LoginSystem.java file the code is taking null values for username and password and opening the application for null values of username and password, try changing the code for that,
5. Back button in the password storage page to go back to the AppGUI not by closing the window.

Refer the documentation of black box, white box test cases and all the docs in the git for getting a broader idea and modify the source code as per the pass and fail cases