

Module 5- Computer Systems (2021-22)

Project

UNIVERSITY OF TWENTE.

Software Testing Document (STD) Template

Team ID: 17	Project Name: ChessMate
Team members: Hein Huijskes, Julia van der Geest, Ujjwal Dodeja, Jose Gavilanes, Maouheb Bessi, Mengmeng Li	Mentor (s): Ramish Bhutto, Wenjie Zhao

Instructions:

1. Refer to the below table and complete all the sections with clarity.
2. Select those test strategies that are applicable to test your application.
3. Make sure to refer to the "Development-Security by Design Checklist" to see the possible vulnerabilities in your application.
4. Feel free to add features and test cases in the table that are essential to test your application.
5. You can use Selenium, SonarQube, and/or GitLab CI/CD to perform source code review, static and dynamic application testing, etc.

Test Strategy	Date (When did you perform the testing?)	Process/Function (Features to be tested)	Test Case	Step	Description	Status (Passed/Failed /Open)	Expected Results	Actual Result	Mitigation plan/Solutions	Review on the Mitigation plan (Passed/Failed)	Remarks on the Failed mitigation plan
Example: Application Testing/API Testing/Acceptance Testing, Manual Testing, etc.		Example 1: Authentication	Example: Logging In	1	Example 1: The correct user ID and password should be entered.	Failed	Example 1: User should access the home page.	User directed to another page.	Example 1: i) Check if there is any connection issues between the home page and its sub page. ii) Check if there is any design issue., iii) Check if this test case is not passed with a particular user, etc.	Passed	...
Application Testing	01-11-2021	SQL	Database content	1	The database should update and fetch information properly.	Passed	The queries should return the results we put in the database before by updating it.	The results were in the database and when fetched, the information is properly returned to the code.	i) Check access to database ii) Make necessary changes in source code.	Not Required	...
Manual Testing	01-11-2021	SQL	SQL Injections	1	The user should not be able to inject code into the database.	Passed	SQL injections should be handled by the database by prepare statements.	The injected code gets handled accordingly by the database and ends up in the table.	i) Use prepare statements in SQL. ii) Filter out user input.	Passed	

Application Testing	01-11-2021	AI	Playing against the computer	2	The AI should make a valid move after the user’s move.	Passed	The AI makes valid moves against the user within 5 seconds of the user’s move.	The AI makes a valid move instantly which is updated on the board.	(i) Make necessary changes in the AI algorithm and the source code (ii)If required and feasible, use a different AI	Not Required	
Manual Testing	01-11-2021	Giving a move	Moves as input	1	The system should ask the player to make his move.	Passed	The system asks the user to give a valid move as input.	The system asks the player to give a move.	(i)Create a prompt which asks for a move (ii)Take a pair of coordinates as input	Not Required	
Application Testing	01-11-2021	Analyse move	Recognising and reflecting a valid move	2	The system should analyse the move input, reflect it on the board and save necessary information.	Passed	The given move should be reflected on the board.	The move was reflected on the board. In case of an invalid move, a prompt message was displayed asking for a new move.	(i) A mapping is created for similar input words (ii) If not easily recognised, the move is compared to the mapping to extract valid coordinates	Passed	

**Note: Refer to the following documentation on GitLab and SonarQube for clarity-**

- Source Code review with SonarQube: <https://docs.sonarqube.org/latest/>
- GitLab integration with SonarQube: <https://docs.sonarqube.org/latest/analysis/gitlab-integration/>
- SonarQube (Static Application Testing):  
<https://www.sonarqube.org/features/security/>
- Gitlab (Static Application Testing):  
[https://docs.gitlab.com/ee/user/application\\_security/sast/](https://docs.gitlab.com/ee/user/application_security/sast/)
- GitLab (Dynamic Application Testing):  
[https://docs.gitlab.com/ee/user/application\\_security/dast/](https://docs.gitlab.com/ee/user/application_security/dast/)

**Prepared by:**  
Dipti K. Sarmah (Project Coordinator)