Assignment 4: Software Testing Document (STD) with Security by Design (SBD)

Project Information

Project Name	Team ID	Mentor(s)
Pi Vision	31	Niek Aukes & Lucas Pruijssers

Table 1: Vulnerabilities and Mitigation Summary

Points (Manual and Automated Testing)	Vulnerabilities Identified	Mitigation Plan	Tick ✔	Remarks
1. **Unit Test 1:** Weak Security in Functions	**SQL Injection:** Using trivial SQL injection queries will grant access to the database of the program.	Upgrade the SQL queries by providing them with prepared statements which will enhance security of the database.	✔	Attempted SQL injection commands through Unit test which will remove a known user from the database: ' OR 1=1; DELETE FROM users WHERE user_name= 'bob'. As expected, the prepared statements introduced in our code worked correctly, protecting the database from being altered.
2. **Unit Test 2:** Proper functionality of signup.	**Invalid credentials:** Without a fully functioning or unstable Signup mechanism, the user would either have to make a new account very frequently, or wouldn't need one at all. This would defeat the purpose of having a personal set of instructions in your personal account. Or personal instructions would be impossible.	By using unit tests we make sure the functionality of the add_user() function is limited to certain constraints making sure the signup process is integrated seamlessly the way it is supposed to.	✔	All unit tests regarding the signup process passed, or gave the expected/ wanted result. They seem to grasp the main functionality the signup process should have. No further remarks.

3. **Manual Security Test 1:** Malicious Plug-In	**Exposed ports vulnerability**: Unused ports will be available for unauthorized access, such as USBs carrying malicious content, which can compromise the Pi.	Acting at Kernel Level and implement a custom BlackList that will only allow the camera to run	✔	The initial solution was to remove power from ports at the application level, which was weak as ports would start again when a component is ejected. A more kernel-oriented approach was then considered, initially going for a WhiteListing approach to finally implement a BlackListing approach. To test it, we inserted a USB Storage Unit to check whether the storage could be recognized by the Raspberry Pi or not. It was not possible to find the USB, meaning the blacklist worked correctly.
4. **Manual Security Test 2:** SQL Injection	Possibility of code insertion aimed at modifying/altering the database structure.	Input sanitation using prepared statements	✔	We have manually tried to bypass the password, using a trivial SQL injection command such as the following: " or “”1 == 1, --”. In this way we tried to enter the webserver without a valid username or password. The actual result is that the query didn’t produce the result specified, meaning that inputs in our database are sanitized and do not allow for injection.
5. **Manual Security Test 3:** Access control Check	**Unauthorized access**: The possibility for a user to access another person's account or the main login page without being logged in	Implement user sessions in order to disallow people to modify and access data which is not theirs.	✔	We have manually tried to modify the URL of the webpage to access user pages while not signed in. It is not possible to access such a page/user unless the correct credentials for that account are inserted.

Table 2: Testing Log (Unit and Manual Tests):

Test Strategy	Date	Process/ Function	Test case step	Description	Status	Expected result	Actual result	Mitigation plan/solution	Review on mitigation plan	Remarks on the failed mitigation plan
Unit Test	29/10/2025	Database sanitization	Attempt SQL Injection using Unit Test	Verify database is secure and safe from any malicious attempt	Passed	No unwanted database query is ran	Only sanitized input and therefore no malicious input	Prepared statements were introduced to prevent users from running commands in the input fields when signing up or logging in.	PASS	X
Unit Test	29/10/2025	Sign up Functionalities	Ensure that valid users are created correctly and assigned a session.	Verify that a user can be created if and only if all criteria for creation are met	Passed	A user is created upon sign up if all criteria are met and a session is created	A user is created if the username is not taken, and not if it is. No user can be created without a username or password	Show an information message	PASS	X
Manual Test	20/10/2025	Exposed ports vulnerability	Attempts to connect a USB-stick to the raspberry Pi	Only allow the camera and not other external element to run on the Raspberry Pi	Passed	No external device plugged in should be recognized other than the camera	It worked as expected. The usb stick did not get recognized but the camera did	Introduce a blacklist so that different peripherals will not get access to the Pi	PASS	X
Manual Test	29/10/2025	Security enhancement	Attempt manual SQL Injection	Verify that the database is	Passed	No data is modified in the	A new user was created	Use prepared statements to	PASS	X

				secure and safe from any malicious attempts		database apart from what should have been added/removed	with the username being the attempted injection script	prevent user input from being interpreted as a command		
Manual Test	29/10/2025	Sessions (not accessing pages that you aren't authorized to)	Attempting to access another user's page without having their credentials	Preventing non-authorized people from accessing data is not theirs	Passed	Users should not be able to access other users accounts without valid credentials, even if the URL is modified.	Changing the URL attempting to access other user's data will be redirected to your user page.	Implement user sessions for every user in order to prevent unauthorized access	PASS	X

Table 3: Usage of AI Tools and Team Review:

Team member name	Reviewed document	Date reviewed	AI tools used	If Yes, Specify how used
Bas Mulder	✔	29/10/2025	Yes	I have used AI tools to generate pictures we're planning on using on our webserver.
Daniel Sazykin	✔	29/10/2025	Yes	I have used AI tools to familiarise myself with new python libraries such as the ones used for GUI and listeners.
Kapllan Sheno	✔	29/10/2025	Yes	I have used AI tools to refine and restructure the wording in the draft. No direct copying of text blocks.
Mirko Troise	✔	29/10/2025	Yes	I have used AI tools to understand the correct and most suitable names for the vulnerabilities we identified.
Paul Mandaag	✔	29/10/2025	Yes	I have used AI tools to check if my implementations of code snippets are correct/up to standards.
Rick Arentsen	✔	29/10/2025	Yes	I have used AI tools to generate ideas for code implementations and review the code.