Error 0:

We wish to note that the setup for team 2 SafeHome is quite complicated and does not work as intended in the ReadMe/UserManual. None of us managed to install all the requirements as intended (with UV and executing “scripts/activate_venv.sh”)

Error 1: Arming with open door/window sensors

Input Sequence:

-Activate the system in the control panel
-Login to the Web App as admin
-Click on “Sensors”
-Write “1” in WinDoor Sensor “Input ID”
-Click “Intrude” in WinDoor Sensor
-Back on the Web App, click “Security” and input password
-Activate all zones (including the one which contains sensor 1)
→ System arms the zone successfully without any warning

Erroneous behaviour:

Based on Team 2’s SRS, in use case 2.b. “Arm/disarm system through web browser”, the exception for step 7 states that when the homeowner chooses to arm the system and the doors or windows to be armed are not closed, the web browser displays the message “Doors and windows are not closed” and the mode does not change. Similarly, in use case c. Arm/disarm safety zone selectively, the exception for step 7 specifies that if the doors or windows to be armed in the selected safety zone are not closed, the system must display the message “Doors and windows are not closed”.

The original SafeHome SRS defines the same rule: in use case b. Arm/disarm system through web browser, exception 7a states that when the homeowner chooses to arm the system and the doors or windows to be armed are not closed, the system displays the message “doors and windows not closed”; and in c. Arm/disarm safety zone selectively, exception 7a requires the same message when doors or windows in that safety zone are not closed.

In the described input sequence, WinDoor sensor 1 is explicitly set to an intrude/open state before arming all zones, including the one containing that sensor. According to both Team 2’s SRS and the original SRS, the system should detect that at least one door/window sensor in an

armed zone is not closed, refuse to arm that zone (or the whole system, depending on the command), and display the “Doors and windows are not closed” message.

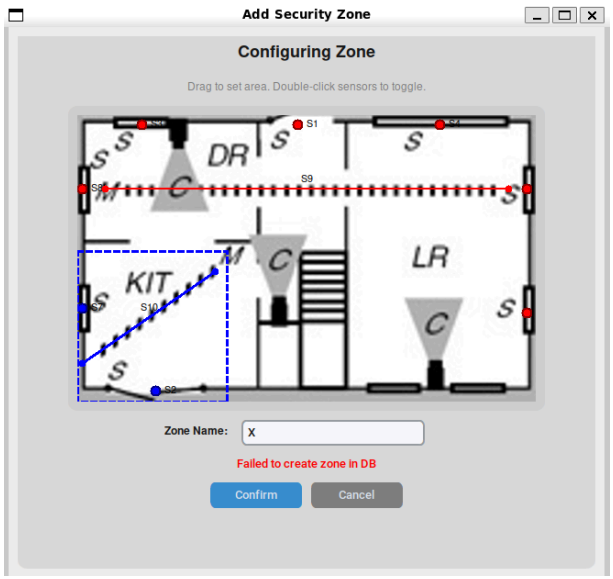
→ However, the implementation arms the zone successfully without any warning, even though WinDoor sensor 1 is in an open/intrusion state. This violates the arming exceptions in Team 2’s SRS (use cases 2.b. Arm/disarm system through web browser and 2.c. Arm/disarm safety zone selectively) and the equivalent use cases in the original SafeHome SRS, so this is a clear SRS/SDS conformance bug.

Error 2: Zone name cannot be reused

Input Sequence:

-Activate the system in the control panel
-Login to the Web App as admin
-Click on “Security” and input the password again.
-Delete the “Kitchen” zone (bottom left).
-Create a new zone (with the same sensors as Kitchen) with name X
-Delete zone X
-Create a new zone with name X
→Zone is not created with message “Failed to create zone in DB”

Resulting Bug:



Erroneous behaviour:

Based on Team 2's SRS, Security Use Case f. Create new safety zone, the system shall save a newly created safety zone in the database and confirm successful creation, with a specific exception only if it fails to save the safety zone in the database (ran out of storage space). Security Use Case g. Delete safety zone specifies that, after the homeowner confirms deletion, the system deletes the safety zone and reports successful deletion. Together, these use cases imply that a safety zone which has been successfully deleted is removed from the database and that a new zone with the same name and sensors can subsequently be created, unless the database is actually unable to save it because of the storage-space error described in the exception.

The original SafeHome SRS (Team 3 from 2007) reinforces this: the Create new safety zone and Delete safety zone use cases state that the system saves new safety zones and, after deletion, saves the remaining safety zones for the system and sensors, indicating a consistent, reusable configuration in storage.

→ In the observed implementation, after deleting the "Kitchen" zone and a newly created zone X, attempting to create zone X again produces the message "Failed to create zone in DB" even though there is no indication of database storage exhaustion or a remaining duplicate zone. This contradicts both Team 2's SRS and the original SafeHome SRS expectations for create/delete behaviour and error conditions, so this is a requirements/design bug.

Error 3: Safety Zone Deleted Without Confirmation Dialog

Input Sequence:

-Activate the system in the control panel
-Login to the Web App as admin
-Click on "Security" and input the password again.
-Select a zone
-Click on "Delete Zone"
→ The system immediately deletes the zone without asking for any confirmation

Erroneous behaviour:

Based on Team 2's SRS, functional requirement section for Security, use case g. Delete safety zone, the basic flow specifies that when the homeowner chooses to delete a safety zone, the system must first prompt the user to confirm the deletion, and only after the homeowner confirms does the system delete the zone from the database and update the configuration. The exception flow for this use case concerns database failures, not the absence of confirmation.

The original SafeHome SRS defines the same interaction pattern in its Delete safety zone use case: after the homeowner selects a zone to delete, the system displays a confirmation dialog and proceeds with deletion only if the homeowner confirms. This makes the confirmation step a mandatory part of the delete operation, to avoid accidental removal of zones and their sensor assignments.

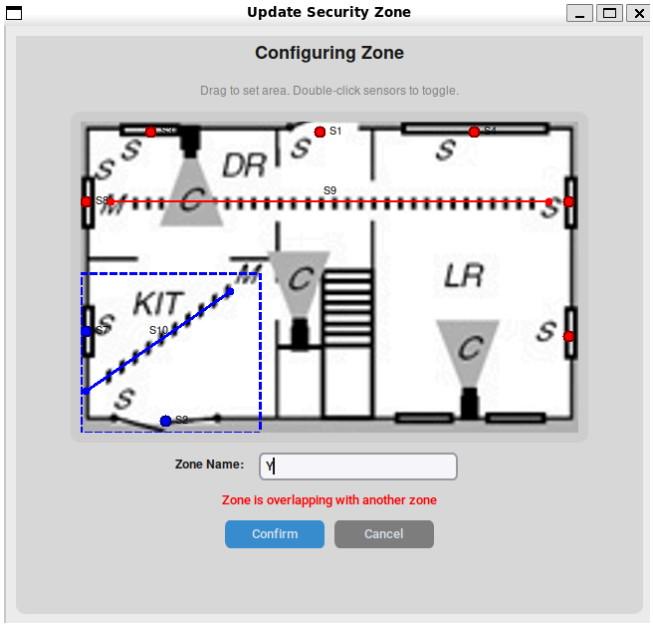
→ In the observed implementation, after selecting a zone on the Security page and clicking Delete Zone, the system deletes the zone immediately without presenting any confirmation dialog. This bypasses the required confirmation step described in Team 2's Delete safety zone use case and in the original SafeHome SRS, and therefore constitutes an SRS/SDS conformance bug.

Error 4: Cannot Update Zone Name - Overlap Check Prevents Renaming

Input Sequence:

-Activate the system in the control panel
-Login to the Web App as admin
-Click on "Security" and input the password again.
-Create a new zone X on the bottom-left (same place as Kitchen)
-Select the X zone
-Click on "Update Zone"
-Change *only* the name to Y
-Click "Confirm"
→ Zone is not changed, message "Zone is overlapping with another zone" appears.

Resulting Bug:



Erroneous behaviour:

Based on Team 2's SRS, Security use case h. Update an existing safety zone states that the homeowner may update one or more details of a safety zone; the system then tests the updated sensor connections and validates the configuration, asks for confirmation, and on confirmation saves the updated configuration to the database and reports success. The only error conditions defined are: invalid or conflicting updates on sensors, a selected sensor already included in another safety zone, or failure to save the safety zone in the database because storage space has run out. There is no requirement or exception that forbids overlapping zones or that rejects a change when only the name of the zone is modified. The same document's open issues list explicitly asks "What if homeowner wants to have overlapping Safety Zones?", indicating that behaviour for overlapping areas is not specified as a constraint. The original SafeHome SRS use case Update an exist safety zone likewise describes selecting an existing safety zone, selecting update, changing its information, and then saving; the only exceptions concern missing selection or alarm conditions, with no mention of overlap checks or geometric conflicts. For creating a new zone, the original SRS defines a separate exception when a new safety zone is the same as an existing one, in which case the message "same safety zone exists" is shown. This check applies to creating a new zone, not to renaming an existing one.

In the described sequence, the homeowner creates zone X in the same physical area as the former Kitchen zone, then uses Update Zone to change only the name from X to Y. The zone's area and sensors remain unchanged; only the descriptive name is edited. According to both SRS versions, this should be a valid update and must be saved successfully.

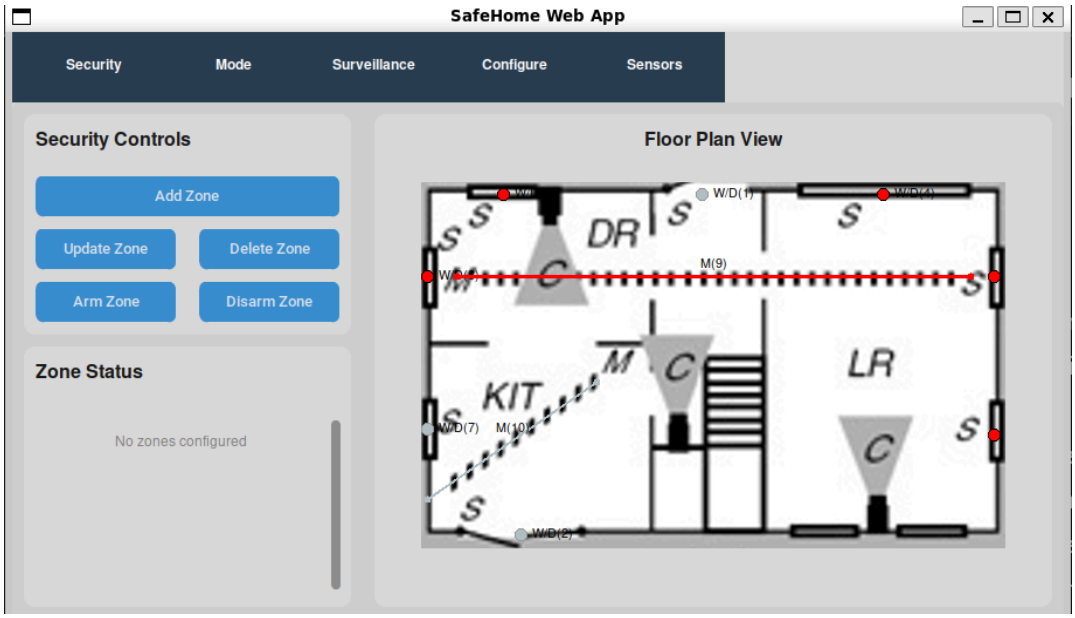
→ However, the implementation rejects the update and displays the message “Zone is overlapping with another zone”, even though no change was made to the zone’s geometry and no such overlap-based rejection is specified in Team 2’s SRS or in the original SafeHome SRS for the update operation. This means the overlap check is incorrectly blocking a pure rename, and the produced error does not correspond to any defined exception, so this is an SRS/SDS conformance bug.

Error 5: Deleted Zone's Sensors Remain Visually Armed

Input Sequence:

-Activate the system in the control panel
-Login to the Web App as admin
-Click on “Security” and input the password again.
-Create a safety zone X with several sensors (or select a previously existing one)
-Select the zone X
- Click on “Arm Zone”
-Delete zone X
→ The sensors are still visually red on the floor plan, ie. marked as armed.

Resulting Bug:



Erroneous behaviour:

Based on Team 2's SRS, Use Case 2.g. Delete safety zone, the system must delete the selected safety zone after the homeowner confirms, and it "deletes the safety zone and displays Successfully deleted Safety Zone {name}." In the original SafeHome SRS swimlane Delete safety zone under Configure safety zone, the step Save safety zones after deleting explicitly states that "The safety zones is saved for system and for sensors." This means that when a safety zone is removed, the system must also update the configuration seen by the sensors and the floor-plan view, so their armed/disarmed status is consistent with the new configuration.

In the observed implementation, after arming safety zone X and then deleting it, the sensors that belonged to X remain shown as red (armed) on the floor plan even though zone X no longer exists and has been deleted from the configuration.

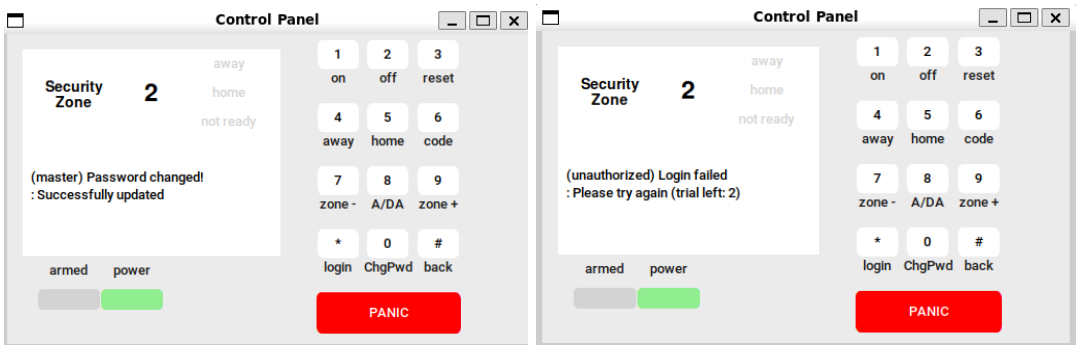
→The visual status of the sensors is therefore inconsistent with the safety-zone configuration that has supposedly been "saved for system and for sensors," violating both Team 2's SRS delete-zone behaviour and the original SafeHome SRS definition of how deletions must propagate to sensors.

Error 6: Change Master Password Does Not Work

Input Sequence:

-Activate the system in the control panel
-Press "6" to enter Function Mode
-Press "0" to change master password
-System displays "Enter new password"
-Enter a 4-digit new password (e.g., "5678")
-System displays "Re-enter new password"
-Re-enter the same 4-digit password (e.g., "5678")
-System displays "Password changed! Successfully updated"
-Press "#" to go back to initialized state
-Attempt to login again using the NEW password "5678"
→ Login fails - the system still requires the OLD password "1234"

Resulting Bug:



Erroneous behaviour:

Based on Team 2's SRS, Common Function use case 1.g. Change master password through control panel specifies that after the homeowner logs onto the system, activates the change-password sequence, enters the current password and then enters and re-enters the new password, the system checks the current password and verifies that the two new passwords match, and then changes the master password. The only defined exceptions are when the current password is not appropriate or when the two new passwords are not the same; in both cases the system asks for re-entry, and the password is not changed. No exception describes a situation where the system announces success but silently keeps the old password.

The original SafeHome SRS defines the same behaviour in its Common Function use case 1.g. Change master password through control panel, where, after the current password is accepted and the new password is entered twice, the system changes the master password. The associated swimlane diagram for "Change Master Password Through Control Panel" shows the same sequence ending with the system changing the password once the new passwords match. In the described sequence, the control panel displays "Password changed! Successfully updated" after the new password is entered and confirmed, which corresponds exactly to the successful end of use case 1.g: the master password is supposed to be updated to the new value. According to both SRS versions, subsequent logins must then accept the new password and no longer accept the old one.

→ However, when attempting to log in again, the new password (e.g. 5678) is rejected and the system still requires the old password (1234). This contradicts the specified outcome of Common Function g. Change master password through control panel in both Team 2's SRS and the original SafeHome SRS, because the system indicates success but does not actually change the stored master password. This is therefore a clear SRS/SDS conformance bug.

Error 7: Password Fail Counter

Input Sequence:

-Activate the system in the control panel
-Press "6" to enter Function Mode
-Press "*" to login.
-Press "*" again to login to master
-Input wrong code (ie. not “1234”)
-System displays “Login failed: please try again (trial left: 2)”
-Press "#" to go back.
-Press "*" to login.
-Press “#” to login to guest
-Press "6" to enter Function Mode
-Press "*" to login.
-Press "*" again to login to master
-Input wrong code (ie. not “1234”)
-System displays “Login failed: please try again (trial left: 2)”
→ System reset the wrong password counter

Erroneous behaviour:

Based on Team 2’s SRS, common use case 1.a Log onto the system through control panel, exception 3b states that when the password is incorrect the system asks for the password again and, if the homeowner/guest enters an incorrect or unrecognizable password three times in a row, the system locks itself for a predefined time. The original SafeHome SRS defines the same behaviour for the use case Log onto the system through control panel, where exception 3a.2 similarly requires the system to lock after three consecutive incorrect password entries.

In the observed implementation, after one failed master login attempt the message shows that two trials remain, but if the user backs out and later retries a wrong master password, the message again shows two trials remaining instead of one.

→The failed-attempt counter is reset by navigating away from the login flow, so the user can avoid ever reaching the three-attempt lockout. This violates the requirement that incorrect passwords be counted “three times in a row” before lockout, and therefore constitutes a defect with respect to both SRS documents.