**1.Jak monitorować poziom wypełnienia kosza?**

Podstawowym zadaniem naszego SmartBin’a miało być monitorowanie poziomu wypełnienia odpadami. Rozwiązanie tego problemu musiało być precyzyjne, ponieważ od niego zależała poprawność danych, które zostaną przesłane do apliakcji. Nasza grupa miała dwa pomysły na zrealizowanie tego rozwiązania.

Pierwszym z nich było użycie kontraktronu MC-38. Elementy czujnika miały zostać umieszczone na klapie oraz ściance śmietnika. Sensor miał wykrywać moment otwarcia kosza korzystając z czasu systemowego (time.time()) oraz zapisywać go w swojej pamięci. Po zamknięciu śmietnika zmienna ta przyjmowałaby wartość None. W momencie gdy aktualny czas różniłby się od zapisanego w zmiennej o kilka minut świadczyłoby o tym że śmietnik pozostaje otwarty, co może świadczyć o tym, że jest przepełniony.

Drugim pomysłem było skorzystanie z ultradźwiękowego czujnika odległości HC-SR04. Miałby on mierzyć odległość między klapą, a poziomem odpadów w śmietniku. W momencie gdy byłaby mniejsza niż kilka cm świadczyłoby to o tym, że kosz jest pełny.

Oba rozwiązania miały swoje zalety i wady. Pierwsze było proste w implementacji, ponieważ było „zerojedynkowe”. Wykorzystywało tylko jeden czujnik podłączony do jednego pina. Prostota ta była jednocześnie wadą tego rozwiązania ponieważ sprawdzało ono tylko domknięcie śmietnika. W sytuacji gdyby użytkownik zostawił niedomknięty śmietnik urządzenie zgłaszałoby stan „pełny” niezależnie od poziomu odpadów.

Drugie rozwiązanie było bardziej dokładne. Czujnik odległości sprawdzał poziom odpadów niezależnie od tego, czy śmietnik był otwarty. Rozwiązanie to też miało wady. Co w sytuacji gdy przy otwartym koszu czujnik zostanie zasłonięty? Mogło do tego dojść na skutek np. ruchu ręką użytkownika, przyklejenia się śmieci do klapy itp. W tej sytuacji śmietnik byłby w stanie „pełny”, niezgodnym ze stanem rzeczywistym.

Połączenie tych dwóch rozwiązań likwidowało wady każdego z nich. Kontaktron umieszczony na brzegu klapy oraz ścianki sprawdzał domknięcie śmietnika. Czujnik odległości umieszczony w centralnym punkcie pod klapą śmietnika mierzył poziom wypełnienia. Dzięki użyciu dwóch czujników stworzyliśmy cztery stany, w których śmietnik może się znajdować:

1. Zamknięty – niewypełniony

W tym stanie zmienna *full == False*, śmietnik jest pusty, wyświetlany jest napis „CZEKAM NA TWOJE SMIECI” oraz żart.

1. Zamknięty – wypełniony

W tym stanie zmienna *full == True*, śmietnik jest wypełniony, wyświetlany jest napis „JESTEM PELNY WYRZUC SMIECI OPOWIEM CI ZART”.

1. Otwarty – niewypełniony

W tym stanie zmienna *full == False*, śmietnik jest otwarty, wyświetlany jest napis „SMIETNIK OTWARTY ZAMKNIJ POKRYWE”.

1. Otwarty – wypełniony

W tym stanie zmienna *full == True*, śmietnik zgłasza błąd czujnika, który powinien zostać sprawdzony przez użytkownika. Wyświetlany jest napis „BLAD ODCZYTU SPRAWDZ CZUJNIK”

Dzięki odpowiedniemu wykorzystaniu tych stanów możemy precyzyjnie określić, co urządzenie powinno przesłać do bazy danych.

**2.W jaki sposób przesyłać dane o stanie śmietnika do użytkownika?**

Samo sprawdzenie poziomu odpadów to za mało. Nasz śmietnik musiał przesyłać informacje o swoim stanie do bazy danych. Pomysły były trzy: kabel USB, Bluetooth, Wi-Fi. Dwa pierwsze szybko upadły, ponieważ do wysyłania informacji wymagały ciągłego połączenia z komputerem. Zaletą wybrania Wi-Fi było to, że informacje mogły być przesyłane do bazy danych bezpośrednio ze śmietnika, bez użycia urządzenia pośredniego. Dzięki temu każdy śmietnik mógł się w każdym momencie łączyć z bazą danych w chmurze.

**3.Jak śmietnik ma się komunikować z użytkownikiem?**

Zdecydowaliśmy, że do komunikacji z użytkownikiem użyjemy wyświetlacza OLED SSD1306. Umożliwiał on wyświetlanie 6 linii tekstu. Każda linia mogła składać się z maksymalnie 16 znaków. Była to wystarczająca długość, aby wyświetlić informacje o śmietniku oraz odpowiednio przygotowany żart.

**4.Skąd brać żarty?**

Jednym z założeń naszego projektu było wyświetlanie żartów, które miały na celu umilenie wyrzucania śmieci użytkownikom. Potrzebowaliśmy bazy z żartami, z których program losowałby dowolny i wyświetlał go użytkownikowi. Początkowo planowaliśmy umieścić listę żartów w pamięci ESP32. Było to proste i skuteczne rozwiązanie. Miało jednak zasadniczą wadę. Aby rozbudować bazę żartów musielibyśmy zaktualizować plik z żartami w każdym śmietniku. Przy założeniu, że z naszego systemu społecznego wyrzucania śmieci korzysta kilkadziesiąt osób, byłoby to żmudne i czasochłonne zajęcie.

Dlatego też zdecydowaliśmy, że lepiej będzie umieścić listę żartów w bazie danych i pobierać stamtąd jeden z nich według wylosowanego przez program umieszczony w pamięci ESP32 ID.

**5.Jak je wyświetlić?**

Gdy mieliśmy już bazę żartów mogliśmy je wyświetlać. Problemem okazał się rozmiar wyświetlacza. Pozwalał on na wyświetlenie tylko 16 znaków w danej linii. Musieliśmy więc podzielić tekst na linie i w takiej formie go przechowywać. Po pobraniu żartu z bazy funkcja *print\_joke* tworzyła z niego listę linii i wyświetlała go na wyświetlaczu.

Zdecydowaliśmy, że żarty będą się aktualizować w momencie gdy śmietnik jest otwarty. Zgodnie z przyjętymi założeniami żart nie jest wtedy wyświetlany. Dzięki temu nowe „suchary” będą pojawiać się wystarczająco często, oraz unikniemy sytuacji, w której żart zmieni się w trakcie, gdy użytkownik chce go przeczytać.

**6.W jaki sposób przechowywać dane?**

Do poprawnego funkcjonowania naszego systemu potrzebowaliśmy miejsca, które umożliwiało przechowywanie danych użytkowników, koszy, żartów oraz ich aktualizowanie w czasie rzeczywistym. Aby każdy śmietnik oraz aplikacja miały dostęp do danych, baza musiała być umieszczona w chmurze.

Zdecydowaliśmy, że użyjemy Google Firebase. Do zalet tego rozwiązania należy zaliczyć edycję bazy danych w czasie rzeczywistym oraz biblioteki do jej obsługi zarówno z poziomu ESP32, jak aplikacji. Dla potrzeb naszego systemu wystarczający był wariant darmowy.

**7.Jak stworzyć aplikację?**

Kolejną ważną decyzją którą musieliśmy podjąć był wybór sposobu w jaki zrealizujemy interakcje użytkowników z naszym systemem. Potrzebowaliśmy rozwiązania, które umożliwi dodawanie nowych użytkowników i koszy, będzie wyświetlało listę koszy, które zostały zapełnione oraz pozwala potwierdzić wyrzucenie kosza przez użytkownika. Prostym i skutecznym rozwiązaniem była aplikacja desktopowa.

Dzięki znajomości biblioteki Qt stworzyliśmy funkcjonalną aplikację łączącą się z bazą danych i umożliwiającą wykonywanie wymienionych wcześniej czynności.

8 **Jak tworzyć użytkowników?**

Osoba, która chce dołączyć do naszej aplikacji musi najpierw stworzyć nowego użytkownika. Po otwarciu aplikacji, na stronie głównej znajduje się przycisk „dodaj użytkownika”. Po jego kliknięciu wyświetla się nowe okno dialogowe umożliwiające wpisanie swoich danych (imię, nazwisko oraz hasło). Po kliknięciu przycisku „ok”, dane użytkownika zostają przetworzone przez aplikację oraz nadany zostaje unikalny numer identyfikacyjny. Następnie wszystkie informacje o użytkowniku zostają przesłane do bazy danych.

Aby zapewnić unikalność numerów indentyfikacyjnych, aplikacja podczas dodawania nowego użytkownika sprawdza jaki najwyższy numer id już istnieje i przypisuje o jeden większy. Jeżeli w bazie danych nie ma jeszcze żadnego użytkownika przypisany zostaje numer 1.

Podczas procesu dodawania użytkownika problemem mogłoby być niewypełnienie, któregoś z wymaganych pól. Jako zabezpieczenie przed taką sytuacją aplikacja, przed dodaniem użytkownika, sprawdza czy wszystkie pola zostały wypełnione i informuje użytkownika jeżeli, któreś z pól jest puste.

9 **Jak dodawać nowe kosze do systemu?**

Na stronie głównej aplikacji znajduje się przycisk „dodaj kosz”, który umożliwia dodanie nowego kosza do systemu. Po jego kliknięciu wyświetla się okno dialogowe umożliwiające wpisanie potrzebnych danych (numer identyfikacyjny, przypisany do posiadanego przez nas kosza oraz numer mieszkania, w którym się znajduje). Po wpisaniu danych i kliknięciu przycisku „ok” aplikacja przetwarza te dane, a następnie przesyła je do bazy danych.

Podobnie jak w procesie dodawania użytkownika, aplikacja sprawdza czy wszystkie pola zostały wypełnione i wyświetla stosowny komunikat jeżeli, któreś pole jest puste.

10 **Jak powiadomić użytkowników o zapełnionym koszu?**

Nasza aplikacja informuje użytkownika o zapełnieniu kosza na dwa sposoby.

Po pierwsze, podczas włączania aplikacji program sprawdza czy w bazie danych znajdują się kosze oznaczone jako zapełnione. Jeżeli takie znajdzie, po włączeniu się aplikacji wyświetla się okno z komunikatem o tym ile koszy jest pełnych. Jeżeli nie ma żadnych zapełnionych koszy okno to się nie pokazuje.

Po drugie, podczas używania aplikacji użytkownik może w łatwy sposób sprawdzić, które kosze są zapełnione, przechodząc na stronę główną. Wyświetlana jest tam lista wszystkich pełnych koszy wraz z informacją w jakim mieszkaniu się znajdują. Za każdym razem, gdy użytkownik wejdzie na stronę główną lista jest aktualizowana.

11 **Jak zgłosić chęć wyrzucenia śmieci?**

Użytkownik chcący zgłosić się do wyrzucania śmieci musi przejść na stronę wyrzuć śmieci (menu -> wyrzuć śmieci). Wyświetlana jest tam lista aktualnie zapełnionych koszy, do których jeszcze nikt się nie zgłosił. Po wybraniu kosza, a następnie wybraniu siebie z listy użytkowników zostajemy poproszeni o wpisanie hasła i jeżeli wpiszemy poprawne hasło zgłoszenie zostaje zapisane.

Poprzez zaimplementowanie konieczności podania hasła podczas zgłaszania się do wyrzucenia śmieci chcieliśmy uniknąć sytuacji, gdy użytkownik zostanie zgłoszony przez innego użytkownika. To zabezpieczenie uniemożliwia podszywanie się pod innych, ponieważ bez podania poprawnego hasła zgłoszenie nie zostanie przyjęte.

12 **Jak potwierdzić wyrzucenie śmieci?**

Po zgłoszeniu się do wyrzucenia śmieci musimy potwierdzić w aplikacji, że to zrobiliśmy. Należy wtedy przejść na stronę potwierdź wyrzucenie (menu -> potwierdź wyrzucenie). Wyświetlana jestem tam aktualna lista śmietników, do których ktoś się zgłosił. Po wybraniu śmietnika z listy, pojawia się informacja o tym kto zgłosił się do wyrzucenia śmieci oraz przycisk „potwierdź”. Po wciśnięciu przycisku program sprawdza w bazie danych czy ten śmietnik został opróżniony. Jeżeli faktycznie śmieci zostały wyrzucone, użytkownik dostaje jeden punkt. Jeżeli śmietnik nadal jest pełny użytkownik nie może potwierdzić wyrzucenia śmieci i dopóki tego nie zrobi nie otrzyma punktu.

Dzięki sprawdzaniu w bazie danych , podczas potwierdzania wyrzucenia śmieci, czy kosz został opróżniony, zabezpieczamy się przed próbami oszustwa i potwierdzaniu wyrzucenia śmieci, wtedy gdy jeszcze nie zostało to zrobione.