White Box Testing

1) syncDisplay() - map.js

This function synchronises data retrieved from the backend with the markers currently displayed on the map.

- If an new element (ie retrieved from backend but not on the map), it is displayed on the map
- If an element is deleted (ie not retrieved from backend but on the map), it is removed from the map
- Otherwise, the contents of the map marker is updated (eg update availability count on carpark marker)

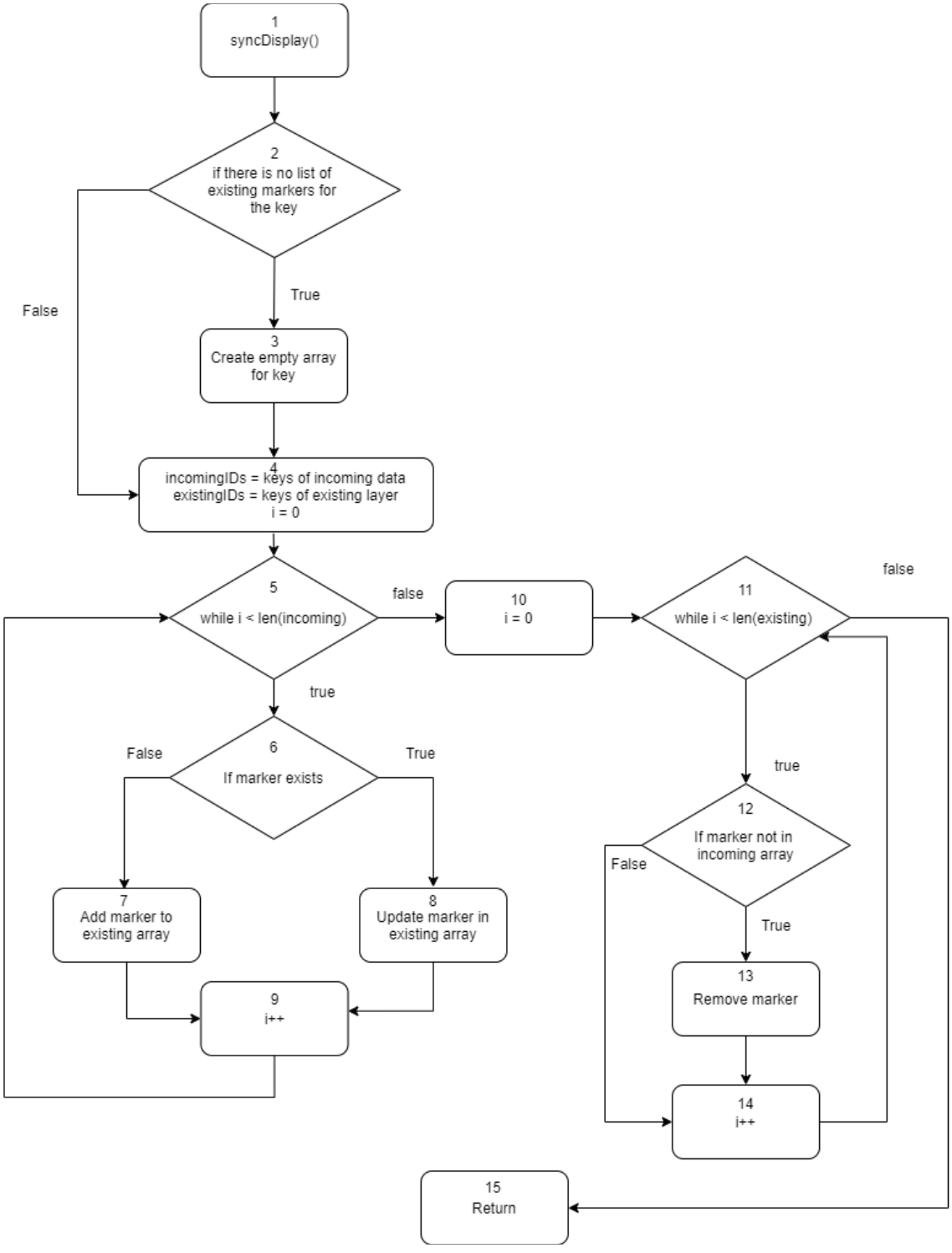
`parentId`: unique identifier to identify which markers to sync to eg “carparks”. Assumed constant, not tested.
`parentLayer`: Leaflet.js layer on which markers are added to. Assumed constant, not tested.
`data`: an array of incoming data from the backend
`display`: a function called to create/update a marker. Assumed constant, not tested.
`displayedLayers`: global object tracking existing markers

```
const displayedLayers = {};
function syncDisplay(parentId, parentLayer, data, display) {
  if (typeof displayedLayers[parentId] === "undefined") {
    displayedLayers[parentId] = {};
  }

  const incomingIds = Object.keys(data);
  const existingIds = Object.keys(displayedLayers[parentId]);

  for (const id of incomingIds) {
    if (!existingIds.includes(id)) {
      // this is a new id, create, then add it to the parent layer
      const layer = display(null, data[id]);
      parentLayer.addLayer(layer);
      displayedLayers[parentId][id] = layer;
    } else {
      // update existing
      display(displayedLayers[parentId][id], data[id]);
    }
  }

  for (const id of existingIds) {
    if (!incomingIds.includes(id)) {
      // id is no longer present, delete
      parentLayer.removeLayer(displayedLayers[parentId][id]);
      delete displayedLayers[parentId][id];
    }
  }
}
```


1
syncDisplay()
2
if there is no list of existing markers for the key
False
True
3
Create empty array for key
4
incomingIDs = keys of incoming data
existingIDs = keys of existing layer
i = 0
5
while i < len(incoming)
false
true
6
If marker exists
False
True
7
Add marker to existing array
8
Update marker in existing array
9
i++
10
i = 0
11
while i < len(existing)
false
true
12
If marker not in incoming array
False
True
13
Remove marker
14
i++
15
Return

Cyclomatic Complexity

CC = |edges| - |nodes| + 2
CC = 19 - 15 + 2 = 6

CC = |decision point| + 1
CC = 5 + 1 = 6

Basic Paths

I. Baseline: 1, 2, 4, 5, 10, 11, 15
II. 1, 2, **3**, 4, 5, 10, 11, 15
III. 1, 2, 4, 5, **6, 7, 9, 5**, 10, 11, 15
IV. 1, 2, 4, 5, **6, 8, 9, 5**, 10, 11, 15
V. 1, 2, 4, 5, **10, 11, 12, 14**, 11, 15 (infeasible - fail to test 12 → 14 without entering first `for`, 12 → 14 is covered in Test case d)
VI. 1, 2, 4, 5, **10, 11, 12, 13, 14**, 11, 15

Test Cases

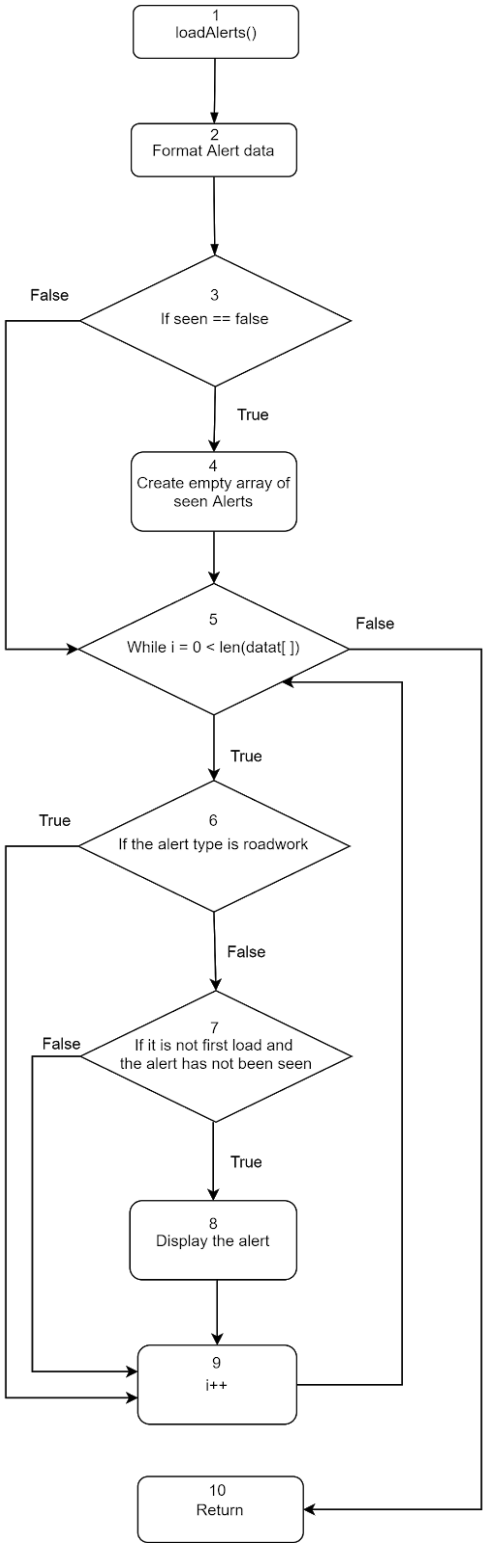
a. Path 1: incoming = [], existing = []
b. Path 2: incoming = [], existing = undefined
c. Path 3: incoming = [`id_1`], existing = [], `id_1` is not in the existing array, add
d. Path 4: incoming = [`id_1`], existing = [`id_1`], `id_1` in both arrays, update
e. Path 6: incoming = [], existing [`id_1`], `id_1` in existing array but not in incoming, delete

Real Execution Paths

a. 1, 2, 4, 5, 10, 11, 15
b. 1, 2, 3, 4, 5, 10, 11, 15
c. 1, 2, 4, 5, 6, 7, 9, 5, 10, 11, 15
d. 1, 2, 4, 5, 6, 8, 9, 5, 10, 11, **12, 14, 11**, 15
e. 1, 2, 4, 5, 10, 11, 12, 13, 14, 11, 15

2) loadAlerts() - manager_alerts.js

This function displays alerts if *not* the first load (to prevent spamming the user with multiple pre-existing alerts), and the alert has not been displayed before. Displayed alerts are saved into a `seen` array.



Cyclomatic Complexity

CC = |edges| - |nodes| + 2
CC = 13 - 10 + 2 = 5

CC = |decision point| + 1
CC = 4 + 1 = 5

Basic Paths

I. Baseline: 1, 2, 3, 5, 10
II. 1, 2, 3, **4**, 5, 10
III. 1, 2, 3, **5, 6, 9**, 5, 10
IV. 1, 2, 3, **5, 6, 7, 9**, 5, 10
V. 1, 2, 3, **5, 6, 7, 8, 9**, 5, 10

Test Cases

I. seenAlertIds = [], data = []
II. seenAlertIds = false, data = []
III. seenAlertIds = [], data = [`id_1`], `id_1` (roadwork) → no display
IV. seenAlertIds = [`id_2`], data = [`id_2`], `id_2` (not roadwork & alert seen) → no display
V. seenAlertIds = [`id_1`], data = [`id_2`], `id_2` (not roadwork & alert not seen) → display

Real Execution Paths

I. 1, 2, 3, 5, 10
II. 1, 2, 3, 4, 5, 10
III. 1, 2, 3, 5, 6, 9, 5, 10
IV. 1, 2, 3, 5, 6, 7, 9, 5, 10
V. 1, 2, 3, 5, 6, 7, 8, 9, 5, 10