CMPE 287 – Software Quality Assurance and Testing

Homework #2

First Name: Harish

Last Name: Marepalli

SJSU ID: 016707314

Professor: Dr. Jerry Gao

Question #1: Basis Path Software Testing (30%)

Based on the given Java program below, please complete the following questions:

A. Generate a control program flow graph for Sort(…) based on the given Java program.(7%)

B. Compute its Cyclomatic number using cyclomatic complexity. (5%)

C. Generate a graph matrix based on your generated flow graph and compute the Cyclomatic metric. (5%)

D. Identify a basis path set (which consists of a number of basis paths) for Sort(….) function. (6%)

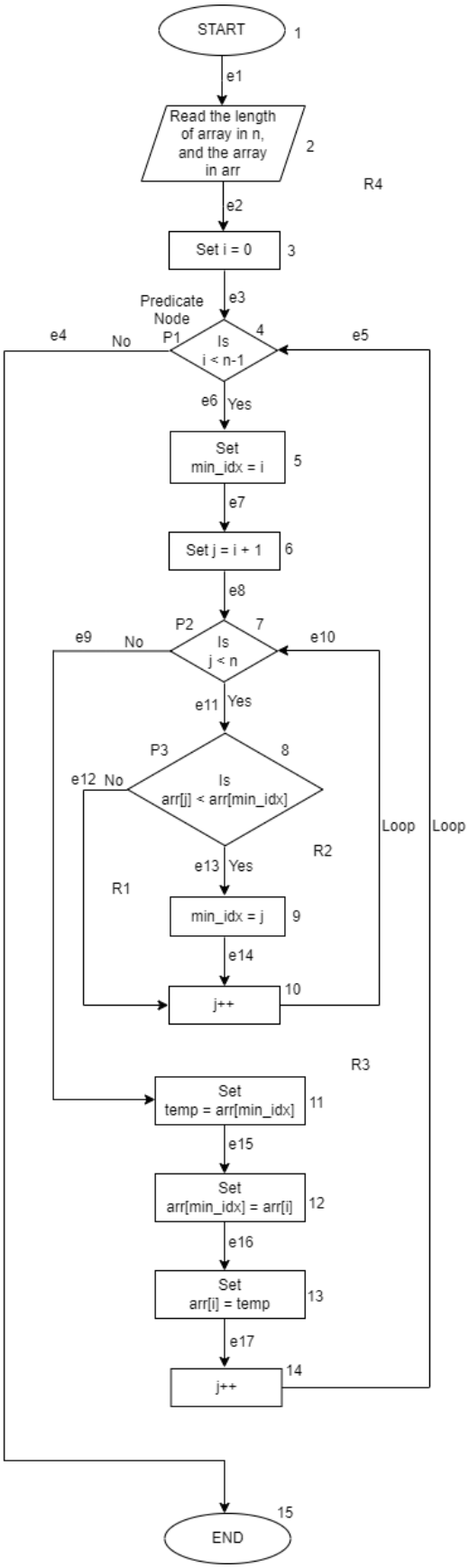
E. List the basis set of test cases (including test inputs and outputs for each test case) (7%)

Java Program to Implement Selection-Sort(….)

https://www.geeksforgeeks.org/selection-sort/

Answer:

A. The flow chart Sort method for the given java program is as follows.


START
1
e1
Read the length of array in n, and the array in arr
2
R4
e2
Set i = 0
3
Predicate Node
e3
P1
Is i < n-1
4
e4
No
e5
e6
Yes
Set min_idx = i
5
e7
Set j = i + 1
6
e8
P2
Is j < n
7
e9
No
e10
e11
Yes
P3
Is arr[j] < arr[min_idx]
8
e12
No
Loop
Loop
R2
e13
Yes
R1
min_idx = j
9
e14
j++
10
R3
Set temp = arr[min_idx]
11
e15
Set arr[min_idx] = arr[i]
12
e16
Set arr[i] = temp
13
e17
j++
14
END
15

B. Cyclomatic Complexity Computation:

M(G) = 4 regions (R1, R2, R3, and R4 from the above flow chart)

M(G) = |E| - |N| + 2 = 17 – 15 + 2 = 4 (In the above flow chart, there are 17 edges indicated as e1, e2, …, e17 and 15 nodes indicated as 1, 2, …, 15)

M(G) = |P| + 1 = 3 + 1 = 4 (In the above flow chart, there are 3 predicate nodes indicated as P1, P2, P3)

C. Graph Matrix:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1-1=0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	2-1=1
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1-1=0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1-1=0
7	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	2-1=1
8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	2-1=1
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1-1=0
10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1-1=0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1-1=0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1-1=0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1-1=0
14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1-1=0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

|P|=3

|P| + 1 = 4

D. Basis Path Set:

Path 1: 1->2->3->4->5->6->7->8->9->10->7->11->12->13->14->4->15

Path 2: 1->2->3->4->5->6->7->8->10->7->11->12->13->14->4->15

Path 3: 1->2->3->4->5->6->7->11->12->13->14->4->15

Path 4: 1->2->3->4->15

So, Cyclomatic complexity = No. of basis paths.

E. Test cases:

For Path 1:

i loop (i<n-1) -> True

j loop (j<n) -> True

if condition (arr[j]<arr[min_idx]) -> False

Input: n = 3 and arr[] = [4, 12, 32]

Actual Output = [4, 12, 32]

Expected output = [4, 12, 32]

Result: Pass

For Path 2:

i loop (i<n-1) -> True

j loop (j<n) -> True

if condition (arr[j]<arr[min_idx]) -> True

Input: n = 3 and arr[] = [32, 12, 4]

Actual Output = [4, 12, 32]

Expected output = [4, 12, 32]

Result: Pass

For Path 3:

i loop (i<n-1) -> False

j loop (j<n) -> N/A

if condition (arr[j]<arr[min_idx]) -> N/A

Input: n = 0 and arr[] = []

Actual Output = []

Expected output = []

Result: Pass

For Path 4:

i loop (i<=n-1) -> True

j loop (j<n) -> False

if condition (arr[j]<arr[min_idx]) -> N/A

Input: n = 1 and arr[] = [20]

Actual Output = [20]

Expected output = [20]

Result: Pass

Question #2: Branch-Based Software Testing (30%)

Branch-Based Software Testing:

A. Based on your generated program flow graph of Sort(….) Function, please generate a branch table. (10%)
B. Generate the identified branch test paths. (10%)
C. List the branch test cases based on your identified paths. (10%)

Answer:

A. Branch decision table:

Based on the above flow chart, the branch decision table is as below. T->True and F->False

Predicate Node	Decision	Possible Outcome
4	i < n-1	T
		F
7	j < n	T
		F
8	arr[j] < arr[min_idx]	T
		F

B. Branch Test Paths:

For this, we create one independent path for each decision's possible outcome.

P->Path

P1: 1->2->3->4->5->6->7->8->10->7->11->12->13->14->4->15

P2: 1->2->3->4->15

P3: 1->2->3->4->5->6->7->11->12->13->14->4->15

P4: 1->2->3->4->5->6->7->8->9->10->7->11->12->13->14->4->15

Decision Table:

Predicate Node	Decision	Possible Outcome	Path
4	i < n-1	T	P1
		F	P2
7	j < n	T	P1
		F	P3
8	arr[j] < arr[min_idx]	T	P4
		F	P1

Predicate Node	Decision	Possible Outcome	Path	T1	T2	T3	T4
4	i < n-1	T	P1	X			
		F	P2		X		
7	j < n	T	P1	X			
		F	P3			X	
8	arr[j] < arr[min_idx]	T	P4				X
		F	P1	X			

C. Branch test cases based on identified paths:

Test Cases:

For Path P1:

i loop (i<n-1) -> True

j loop (j<n) -> True

if condition (arr[j]<arr[min_idx]) -> False

Input: n = 3 and arr[] = [4, 12, 32]

Actual Output = [4, 12, 32]

Expected output = [4, 12, 32]

Result: Pass

For Path P2:

i loop (i<n-1) -> False

j loop (j<n) -> N/A

if condition (arr[j]<arr[min_idx]) -> N/A

Input: n = 0 and arr[] = []

Actual Output = []

Expected output = []

Result: Pass

For Path P3:

i loop (i<=n-1) -> True

j loop (j<n) -> False

if condition (arr[j]<arr[min_idx]) -> N/A

Input: n = 1 and arr[] = [20]

Actual Output = [20]

Expected output = [20]

Result: Pass

For Path P4:

i loop (i<n-1) -> True

j loop (j<n) -> True

if condition (arr[j]<arr[min_idx]) -> True

Input: n = 3 and arr[] = [32, 12, 4]

Actual Output = [4, 12, 32]

Expected output = [4, 12, 32]

Result: Pass

Question #3 Questions about Software State-based Testing (20%).

Figure 1 shows a state diagram for an under-test software feature extraction.

Please answer the following questions.

(a) Generate a state-based test tree based on given Figure 1. (where "idle" is the initial state) (8%)

(b) Identify and list state-based paths for testing in terms of lengths for your test case design. (6%)

(c) Please identify and list the state testing paths from "idle" to "handoff": (6%) (Please avoid the redundant paths)

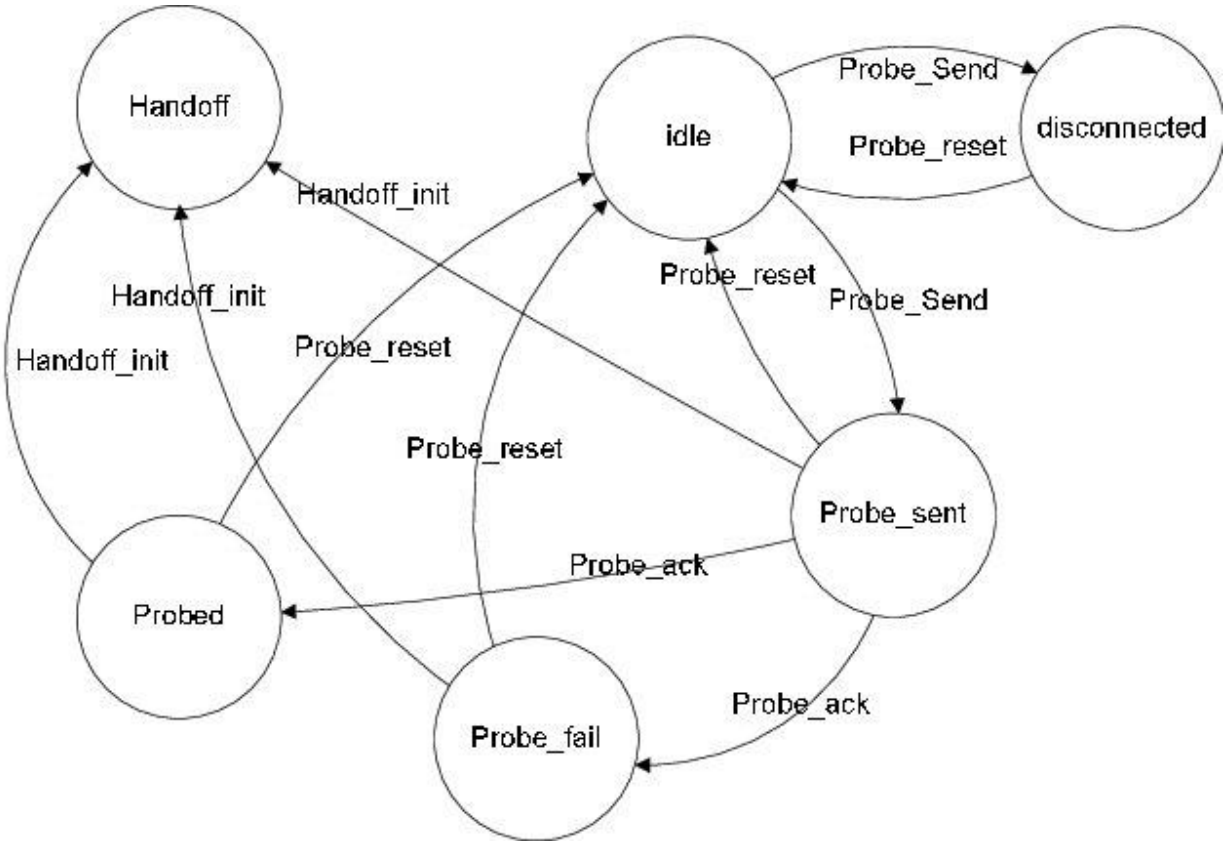
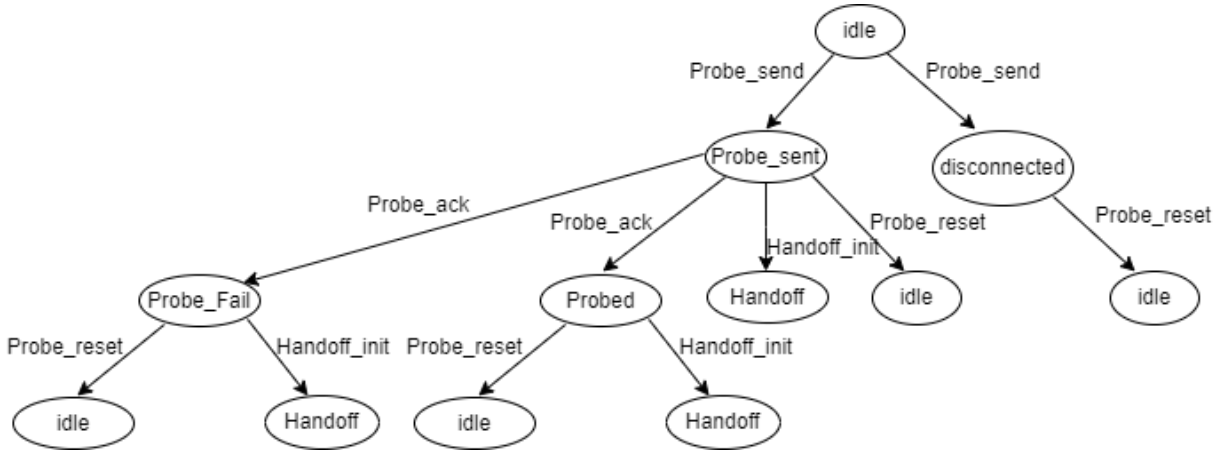


Figure 1 The state diagram

Answer:

(a) State-based test tree:



(b) State-based paths:

Test cases:

Level #1: idle->Probe_send->Probe_sent

idle->Probe_send->disconnected

Level #2: idle->Probe_send->Probe_sent->Probe_ack->Probe_Fail

idle->Probe_send->Probe_sent->Probe_ack->Probed

idle->Probe_send->Probe_sent->Handoff_init->Handoff

idle->Probe_send->Probe_sent->Probe_reset->idle

idle->Probe_send->disconnected->Probe_reset->idle

Level #3: idle->Probe_send->Probe_sent->Probe_ack->Probe_Fail->Probe_reset->idle

idle->Probe_send->Probe_sent->Probe_ack->Probe_Fail->Handoff_init->Handoff

idle->Probe_send->Probe_sent->Probe_ack->Probed->Probe_reset->idle

idle->Probe_send->Probe_sent->Probe_ack->Probed->Handoff_init->Handoff

(c) State testing paths from “idle” to “handoff”:

Below are the state testing paths from “idle” to “handoff”.

Path 1: idle->Probe_send->Probe_sent->Handoff_init->Handoff

Path 2: idle->Probe_send->Probe_sent->Probe_ack->Probe_Fail->Handoff_init->Handoff

Path 3: idle->Probe_send->Probe_sent->Probe_ack->Probed->Handoff_init->Handoff

Question #4: Driverless Car Validation (20%)

Please search the google map with images to identify and discover your answer to the following questions for AV Testing context modeling.

a) Develop and present a road context model for a selected 4-way intersection with traffic lights and cross-walking with meters. (6%)

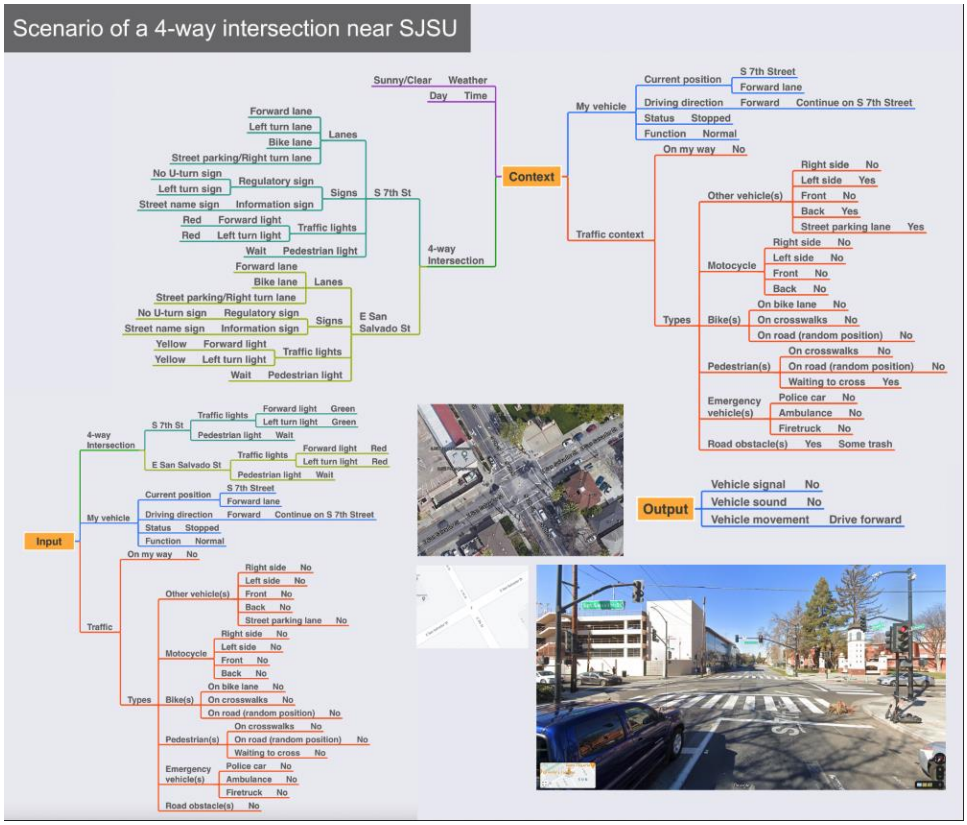
b) Develop and present a road context model for diverse road hazards (based on a well-classified classes) (4%)

c) Develop and present a road context model for people crossing street scenarios (based on a well-classified classes). (6%)

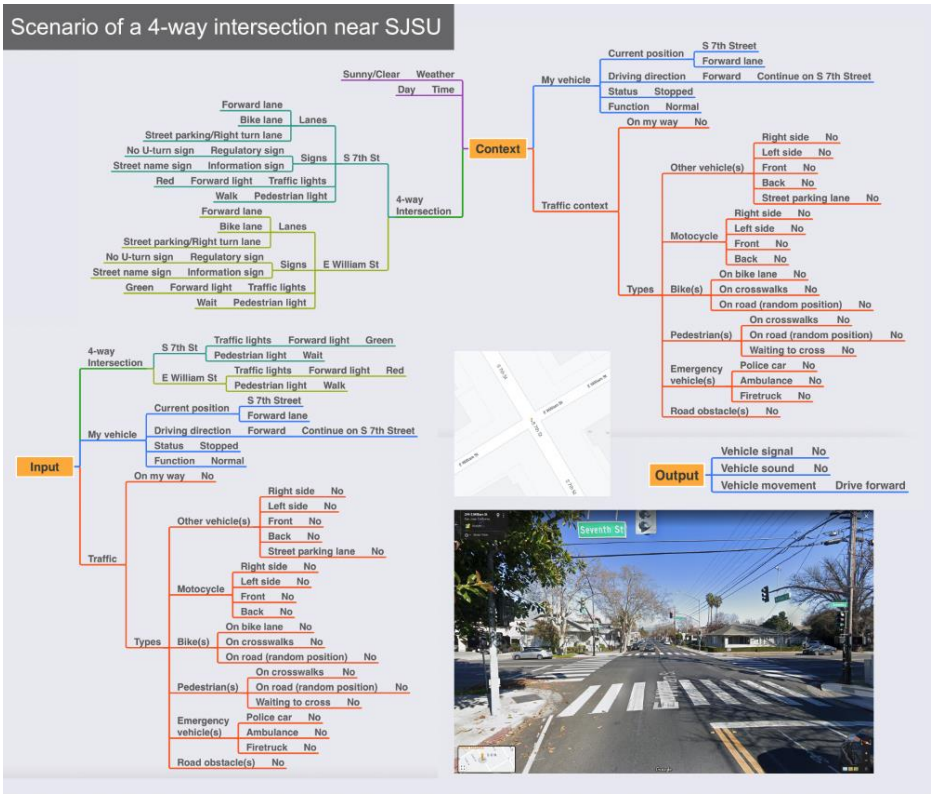
d) Develop a road context model for a school bus scenario (based on a well-classified classes). (4%)

Answer:

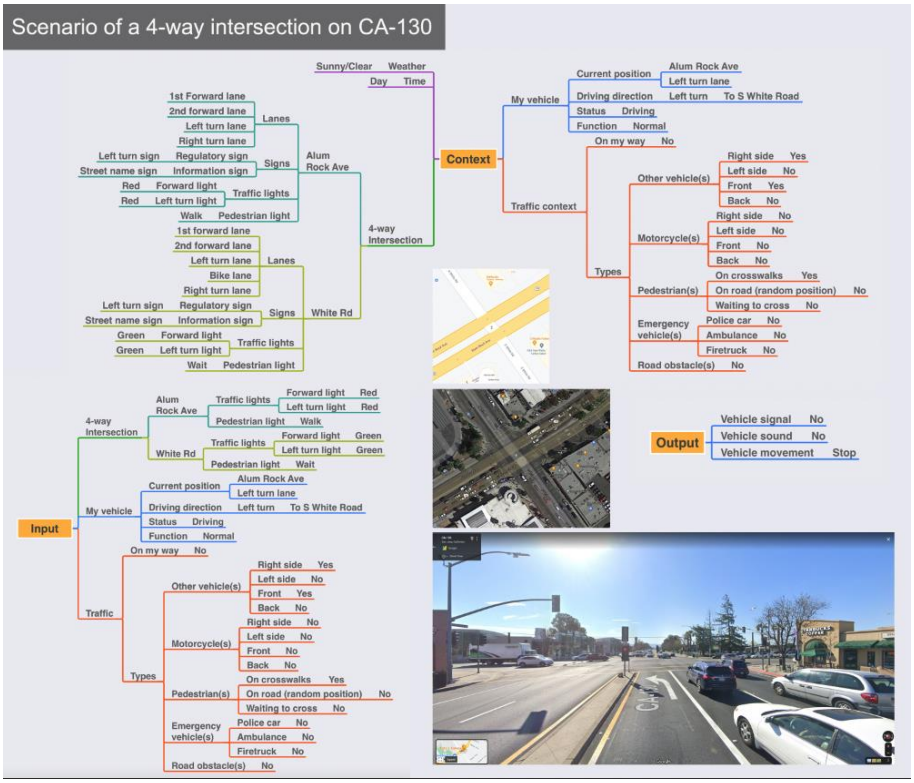
a) Road context model for a selected 4-way intersection with traffic lights and cross-walking with meters is as below.



Scenario of a 4-way intersection near SJSU

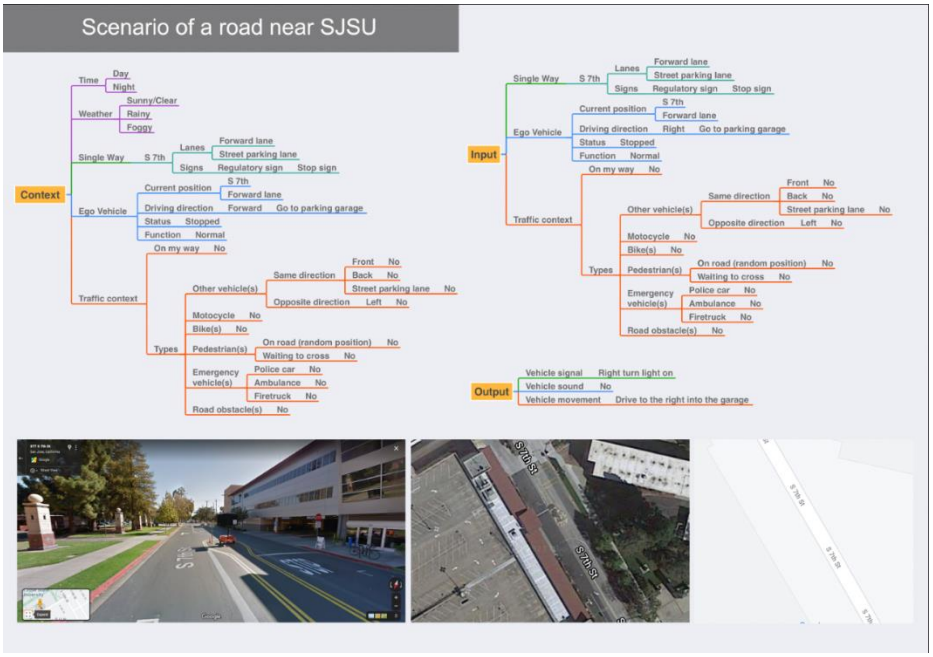


Scenario of a 4-way intersection in East Bay



b) Road context model for diverse road hazards is as below.

c) Road context model for people crossing street scenarios is as below.



d) Road context model for a school bus scenario is as below.