Wildcat Integrated Mobility Solution (WIMS)

Project Report – Team 6

ECE 579

May 1, 2024

Table of Contents

1. Team Members

- Alan Manuel Loreto Cornidez (aloretocornidez)
- Elliot Zeurcher (ezuercher)
- Eric Teitelbaum (ericteitelbaum)
- Ted Ha (ttha)

2. Introduction

The information age was supposed to revolutionize how humans make decisions. With more data at their disposal, people would theoretically become capable of making more informed choices throughout their day, such as how best to travel from one point to another. Instead of making this decision process easier, however, unlimited data has predominantly led to the growing phenomenon of "choice paralysis," the process by which individuals grappling with an overwhelming array of options ultimately end up in a state of indecision. The Wildcat Integrated Mobility Solution (WIMS) serves to remedy this issue by autonomously sorting through a sea of transportation options and making decisions optimized for time or distance, based on user preferences. Unlike common map apps like Google Maps, WIMS offers dynamic guidance on when to change modes of transportation, allowing users to seamlessly transition between different modes throughout their journey. Google maps primarily focuses on setting a route for a single transportation mode, leaving the user to wonder if a better option exists that they are missing. WIMS overcomes this limitation by integrating various transportation options into a single, cohesive travel plan that guarantees an optimal trip.

3. Scope

The WIMS is a tool that could be expanded to navigate any environment with any variety of modes of transportation; however, this project implements a prototype that is limited in scope to the University of Arizona campus and three modes of transportation: walking, biking, or riding the CatTran. For purposes of mimicking a public transit system, the bicycles are assumed to be available for checkout and return at special locations called bicycle depots. Furthermore, the bicycles are assumed to be capable of autonomous navigation in order to demonstrate the functionality of an AI Production System (AIPS) that automatically redistributes the inventory of bicycles between depots as needed. To build the prototype, we constructed a map of the university campus consisting of distinct locations and valid paths connecting them. Locations can be classified as destinations, CatTran stops, bicycle depots, or regular intersections. Destinations are locations where travelers may begin or end their journey. CatTran stops are locations where travelers may board or deboard the CatTran, and bicycle depots are locations where travelers may mount or dismount bicycles. Bicycles are assumed to be available at any time, but the CatTran runs on a specific schedule. Intersections are simply locations where the traveler must choose a new path to continue their journey. These paths can be classified according to the valid modes for traversing them. In addition, we developed our prototype of the WIMS to take into account the effect of traffic conditions on paths and modes of transportation, both spatial and temporal. Although the prototype for the WIMS is limited in scope to the University of Arizona campus, it is intended to be scalable to any complex transportation network. In fact, each of the locations, paths, and transportation options included in the prototype are fully configurable and could be used to mimic any number of real-world environments; for instance, commuting in downtown New York city. In addition, the scope of WIMS could be expanded to include modes of transportation with increasingly complex schedules, such as plane flights or car rentals. Finally, additional measures of merit might be introduced for WIMS to optimize for, such as the monetary cost of travel.

4. Requirements

The project requirements are as follows:

1. WIMS shall read static database files that correspond to locations, paths, and traffic information.
2. WIMS shall take the data read from the database files and consolidate the information to form a graph that represents the possible paths an individual may take to travel across various locations at the University of Arizona.
3. WIMS shall allow the user to specify two locations that correspond to locations on a map of the university.
4. WIMS shall allow the user to specify what to optimize the path search for. The parameters that can be specified are to minimize distance traveled, minimize the time taken to travel between two nodes, or to minimize the time taken to travel between two nodes while taking path traffic into account.
5. WIMS shall use the inputs given by the user to conduct an A-Star search to find the optimal path between the two locations entered by the user.
6. WIMS shall display the path found by the A-Star search algorithm. The display shall contain the following:
    a. The node IDs for the travel path
    b. The order the nodes are taken in the path
    c. The travel method used to arrive at the node.
    d. The performance metric used to measure the optimality of the path.

5. Wildcat Smart Guide

The Wildcat Smart Guide (WSG) is an integrated mobility solution that provides users the smart navigation guide between different places in the University of Arizona campus integrating the time of day, real-time traffic condition and mode of transportation. The amount of people on campus changes throughout the day; The amount of traffic spikes between classes; Certain parts of the campus is more crowded; Certain modes of transportation cannot travel all paths. Considering all such information, the WSG provides the best paths to travel from one place to another place on campus using the best modes of transportation. For example, the WSG may advise you to walk from the Main Library to the Southeast corner of the 2nd Street Garage, then switch to the bicycle all the way to the ECE building.

a. Location and Path Data

Total 59 different locations and 97 different paths on campus are chosen for the simulation. For each location, its latitude and longitude information are collected from the Google Map, as well as information like the types of the location (e.g., destination, bicycle depot, Cat Tran stop) and other neighboring locations. For each path, its path and straight distances are collected using the Google Map, along with information like allowed mode of transportation and the traffic rating, which is described more in detail in the Traffic Simulation section.

The list of destination locations selected for the WIMS are:

- Park Avenue Garage
- Aerospace and Mechanical Engineering (AME) Building
- Electrical and Computer Engineering (ECE) Building
- Education Building
- Old Main
- Main Library
- McKale Center

The list of bicycle depot locations selected for the WIMS are:

- East Side of Park Avenue Garage
- East Side of AME Building
- South Side of ECE Building
- Southwest side of Education Building
- East Side of Old Main
- Northeast Side of Main Library
- West Side of McKale Center

The list of CatTran stops locations selected for the WIMS are:

- 1st and Cherry
- Speedway and Mountain
- 2nd and Cherry
- 2nd and Park
- Old Main Circle
- 4th and Cherry

The University of Arizona campus map used for the WIMS is shown in **Figure 1** below. The legend for the map is described in **Table 1**.

Table 1: WIMS Univerity of Arizona Campus Map Legend

Location Color	Location Type

Path Color	Path Type

Pink		Destination
Orange		CatTran Stop
Yello		Bicycle Depot
Black		Intersection

Red		Walk, Bicycle, CatTran
Blue		Walk, Bicycle
Green		Walk

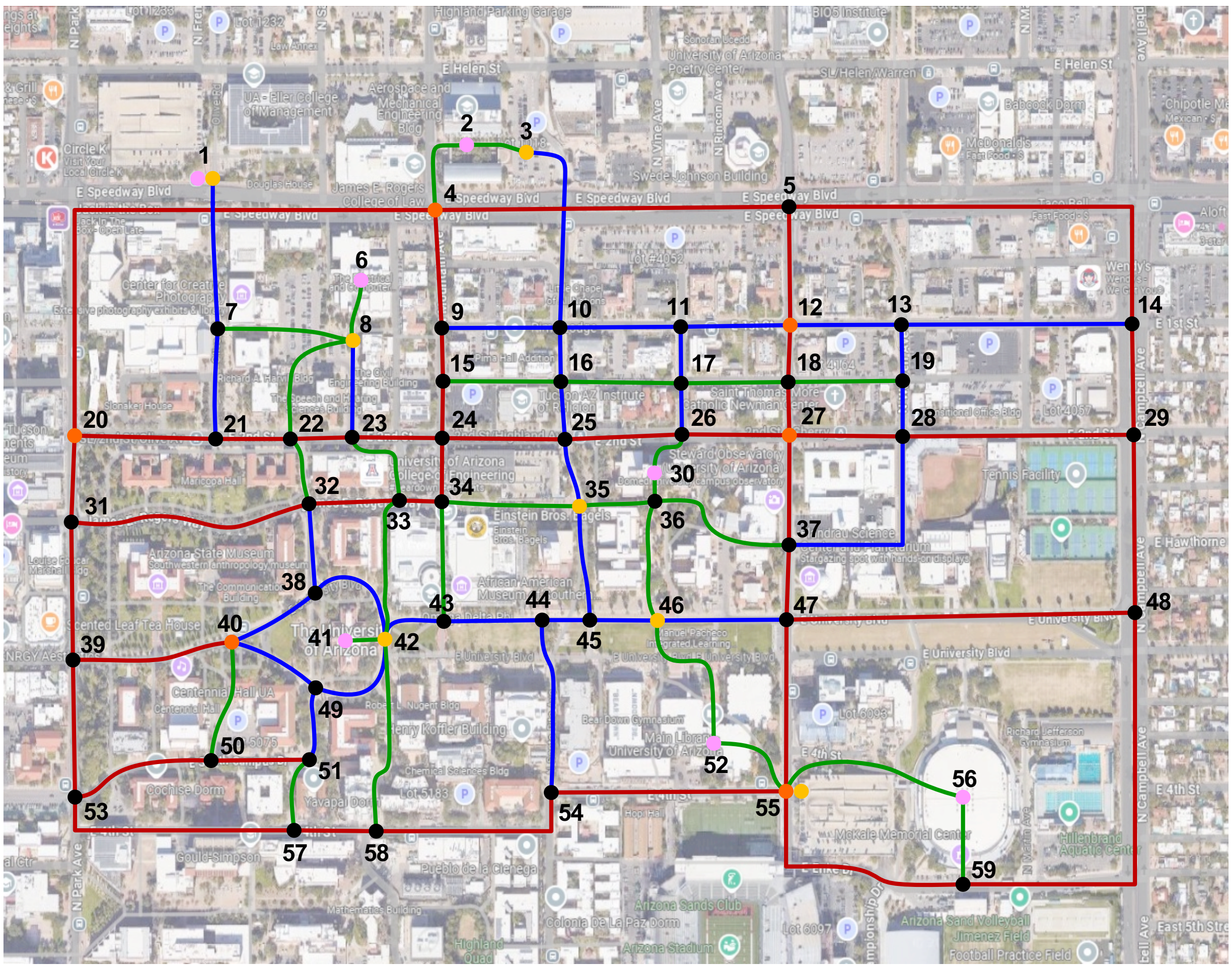


Figure 1: University of Arizona Map for the WIMS

b. Traffic Simulation

The effect of traffic is considered in three different ways: temporally, spatially and by different modes of transportation.

Temporally, the amount of traffic increases in the morning, peaks around noon and decreases in the afternoon. It also spikes between classes. The overall traffic throughout the day is modeled using the Gaussian distribution, where the mean is at 12:30pm, and the standard deviation is 150 minutes. The traffic is simulated from 7:00am to 6:00pm. Assuming the Monday/Wednesday/Friday schedule, the pikes between classes are modeled using the shifted some of the Gaussian distribution, where the mean is at 55min before every hour, and the standard deviation is set to 2 minutes. For the beginning of the first class and for the end of the last class, since students have more time, the traffic is modeled using the Gaussian distribution with the standard deviation of 10 min. The combined temporal modeling of the traffic is shown in the figure below:

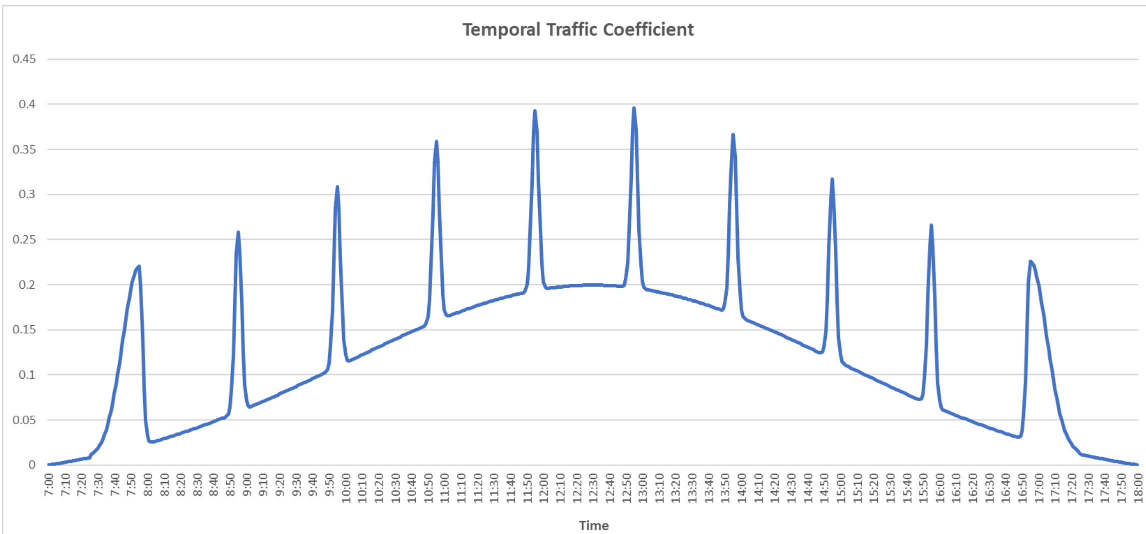


Figure 2: Temporal Traffic Coefficient

Spatially, there is more traffic near the center of the campus where popular classroom buildings and common places (i.e., student union) are more concentrated together. To address this, different paths in the map are given different traffic ratings. There are five traffic ratings, and each rating are given the following traffic spatial coefficient:

Table 2: Spatial Traffic Coefficient

Rating	Color	Spatial Coefficient
5		2.5
4		2.05
3		1.6
2		1.15
1		0.7

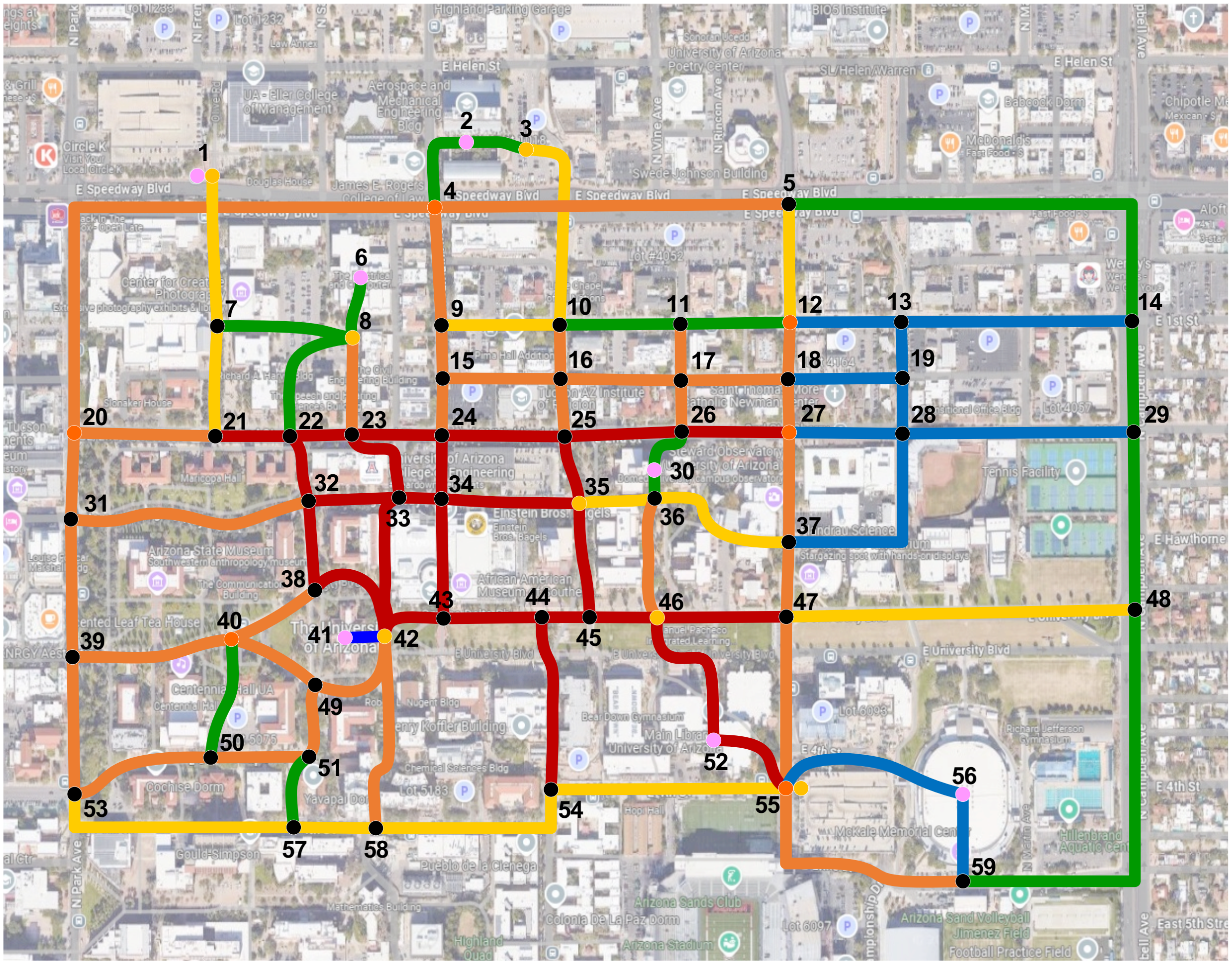


Figure 3: Path Traffic Rating

Also, different modes of transportation are affected differently by the traffic. Walking is not affected by the traffic too much, whereas Cat Trans can be slowed down by more than 70% due to the traffic on campus. The following transportation mode traffic coefficients are used:

Table 3: Modal Traffic Coefficient

Mode	Modal Coefficient
Walking	0.9
Bicycle	0.7
Cat Tran	0.3

Combining temporal, spatial and modal traffic coefficients, the following formula is used to model the traffic:

$$T = 1 - T_t \cdot T_s \cdot (1 - T_m)$$

where T is the combined traffic coefficient that is used to scale the speed of the mode of transportation, T_t is the temporal traffic coefficient, T_s is the spatial traffic coefficient, and T_m is the modal traffic coefficient. The combined traffic coefficient for each mode of transportation is shown in figures below:

Figure 4: Combined Traffic Coefficient for Different Mode of Transportation

c. CatTran Simulation

Analyzing Green and Purple Cat Tran line schedule, the average speed of Cat Tran around the campus is around 11.4 km/h. For this project, the Cat Tran speed is assumed to be 12 km/h. We came up with six Cat Tran stops around the project map, and give the following bus schedule:

Table 4: WIMS CatTran Schedule

12	27	55	40	20	4
7:00	7:01	7:03	7:10	7:13	7:17
7:20	7:21	7:23	7:30	7:33	7:37
7:40	7:41	7:43	7:50	7:53	7:57
8:00	8:01	8:03	8:10	8:13	8:17
8:20	8:21	8:23	8:30	8:33	8:37
8:40	8:41	8:43	8:50	8:53	8:57
9:00	9:01	9:03	9:10	9:13	9:17
9:20	9:21	9:23	9:30	9:33	9:37
9:40	9:41	9:43	9:50	9:53	9:57
10:00	10:01	10:03	10:10	10:13	10:17
10:20	10:21	10:23	10:30	10:33	10:37
10:40	10:41	10:43	10:50	10:53	10:57
11:00	11:01	11:03	11:10	11:13	11:17
11:20	11:21	11:23	11:30	11:33	11:37
11:40	11:41	11:43	11:50	11:53	11:57
12:00	12:01	12:03	12:10	12:13	12:17
12:20	12:21	12:23	12:30	12:33	12:37
12:40	12:41	12:43	12:50	12:53	12:57

13:00	13:01	13:03	13:10	13:13	13:17
13:20	13:21	13:23	13:30	13:33	13:37
13:40	13:41	13:43	13:50	13:53	13:57
14:00	14:01	14:03	14:10	14:13	14:17
14:20	14:21	14:23	14:30	14:33	14:37
14:40	14:41	14:43	14:50	14:53	14:57
15:00	15:01	15:03	15:10	15:13	15:17
15:20	15:21	15:23	15:30	15:33	15:37
15:40	15:41	15:43	15:50	15:53	15:57
16:00	16:01	16:03	16:10	16:13	16:17
16:20	16:21	16:23	16:30	16:33	16:37
16:40	16:41	16:43	16:50	16:53	16:57
17:00	17:01	17:03	17:10	17:13	17:17
17:20	17:21	17:23	17:30	17:33	17:37
17:40	17:41	17:43	17:50	17:53	17:57
18:00	18:01	18:03	18:10	18:13	18:17

For the purpose of the finding the optimal path, when we are optimizing for travel time, the bus schedule is converted to the number of minutes to wait at the node to travel using Cat Tran. The example is shown in the table below:

Table 5: WIMS CatTran Minutes before Next CatTran

Time	12	27	55	40	20	4
7:00	0:00	0:01	0:03	0:10	0:13	0:17
7:01	0:19	0:00	0:02	0:09	0:12	0:16
7:02	0:18	0:19	0:01	0:08	0:11	0:15
7:03	0:17	0:18	0:00	0:07	0:10	0:14
7:04	0:16	0:17	0:19	0:06	0:09	0:13
7:05	0:15	0:16	0:18	0:05	0:08	0:12
7:06	0:14	0:15	0:17	0:04	0:07	0:11
7:07	0:13	0:14	0:16	0:03	0:06	0:10
7:08	0:12	0:13	0:15	0:02	0:05	0:09
7:09	0:11	0:12	0:14	0:01	0:04	0:08
7:10	0:10	0:11	0:13	0:00	0:03	0:07
7:11	0:09	0:10	0:12	0:19	0:02	0:06
7:12	0:08	0:09	0:11	0:18	0:01	0:05
7:13	0:07	0:08	0:10	0:17	0:00	0:04
7:14	0:06	0:07	0:09	0:16	0:19	0:03
7:15	0:05	0:06	0:08	0:15	0:18	0:02
7:16	0:04	0:05	0:07	0:14	0:17	0:01
7:17	0:03	0:04	0:06	0:13	0:16	0:00
7:18	0:02	0:03	0:05	0:12	0:15	0:19
7:19	0:01	0:02	0:04	0:11	0:14	0:18
7:20	0:00	0:01	0:03	0:10	0:13	0:17
7:21	0:19	0:00	0:02	0:10	0:12	0:16
7:22	0:18	0:19	0:01	0:09	0:11	0:15
7:23	0:17	0:18	0:00	0:08	0:10	0:14
7:24	0:16	0:17	0:19	0:07	0:09	0:13
7:25	0:15	0:16	0:18	0:06	0:08	0:12
7:26	0:14	0:15	0:17	0:05	0:07	0:11
7:27	0:13	0:14	0:16	0:04	0:06	0:10
7:28	0:12	0:13	0:15	0:03	0:05	0:09
7:29	0:11	0:12	0:14	0:02	0:04	0:08
7:30	0:10	0:11	0:13	0:01	0:03	0:07
7:31	0:09	0:10	0:12	0:00	0:02	0:06
7:32	0:08	0:09	0:11	0:19	0:01	0:05
7:33	0:07	0:08	0:10	0:18	0:00	0:04

d. Algorithm (A)*

The A* algorithm is applied to find the optimal path. The user can choose either "Distance," "Time with No Traffic" or "Time with Traffic" as the constraint of optimization, since "Distance" and "Time with No Traffic" options are trivial, "Time with Traffic" option is described in detail.

Nodes for the A* algorithm are defined by the Location ID from the map and the mode of transportation that the user took to get there. For example, since there are three different modes of transportation to take to get to Location ID 4, it generates three nodes for the A* algorithm: 4-Walk, 4-Bike and 4-CatTran. In another example, Location ID 7 generates two A* nodes: 7-Walk and 7-Bike. Destination nodes are special since you can start from that node, they also generate an extra "None" A* node. For example, Location ID 1 generates 1-None, 1-Walk and 1-Bike.

Different edges for different modes of transportation are generated for the A* algorithm. For example, between Location IDs 1 and 7, two edges are generated, one for walking and one for biking. The cost for the edge changes constantly depending on the traffic condition and the Cat Tran schedule. The traffic coefficient is multiplied to the speed of the mode of transportation, so that its slower depending on the traffic. For Cat Tran, the amount of time the user needs to wait for the bus is added to the cost for that edge. When a new node is reached in the A* algorithm, the arrival time is recorded so that the correct traffic coefficient and the Cat Tran wait time is used. The speed of each mode of transportation is listed in the table below:

Table 6: Speed of Different Modes of Transportation

Mode	Speed
Walking	5 km/h
Bicycle	16 km/h
Cat Tran	12 km/h

The user can switch the mode of transportation only if the node is a Bicycle Depot or a Cat Tran stop. The user can switch to biking at the Bicycle Depot, but the lock/unlock cost of 1 minute is added when switching to and from biking. At Cat Tran stops, the user can switch to and from Cat Tran to other modes of transportation. When switching from Cat Tran, there is no effect on the edge cost, but when switching to Cat Tran, the wait time is added to the edge cost. Switching between Cat Tran and Bicycle is only possible if the node is both Cat Tran stop and Bicycle Depot (e.g. Location ID 55).

Since the cost of edges change depending on time, source Location ID, destination Location ID, optimization mode and departure time must be given to the A* algorithm.

The heuristic used for the A* algorithm is the straight distance between the current node and the destination. The straight distance is calculated using the following formula:

$$d = R \cdot \cos^{-1}(\sin\theta_1 \cdot \sin\theta_2 + \cos\theta_1 \cdot \cos\theta_2 \cdot \cos(\varphi_1 - \varphi_2))$$

where d is the straight distance between two nodes, θ_1 and φ_1 are latitude and longitude of the first node, θ_2 and φ_2 are latitude and longitude of the second node, and R is the radius of the Earth, which is 6,372.828 km in Tucson.

6. Bicycle Inventory Management

The Bicycle Inventory Management (BIM) is an Artificial Intelligence Production System (AIPS) that manages the inventory of bicycles in bicycle depots depending on the demand during the day using the "autonomous" bicycle that can travel between bicycle depots. The goal of the BIM is that bicycles are always available at bicycle depots and nobody has to wait for a bicycle so that the WSG works accurately and efficiently.

a. Database

The database of the BIMS AIPS is defined by the number of expected bicycles at time $\boldsymbol{t}$ $(N_e(t))$, the actual number of bicycles at time $\boldsymbol{t}$ $(N_a(t))$, and time $\boldsymbol{t}$:

$$(N_e(t), N_a(t), t)$$

b. Rules

The rules for the BIMS AIPS are:

1) Receive a reservation for a bicycle
2) Expect to receive a bicycle from another Bicycle Depot
3) Send a bicycle to another Bicycle Depot
4) Release a bicycle to the user
5) Remove existing reservation
6) Receive a bicycle returned from the user

c. Control Strategy

The control strategies for the BIMS AIPS are described in flow charts. The control strategies for when the user requests a bicycle are described in the figure below. There are two cases to consider, 1) Bicycle is available in the current depot, and 2) Bicycle needs to be requested from another depot.

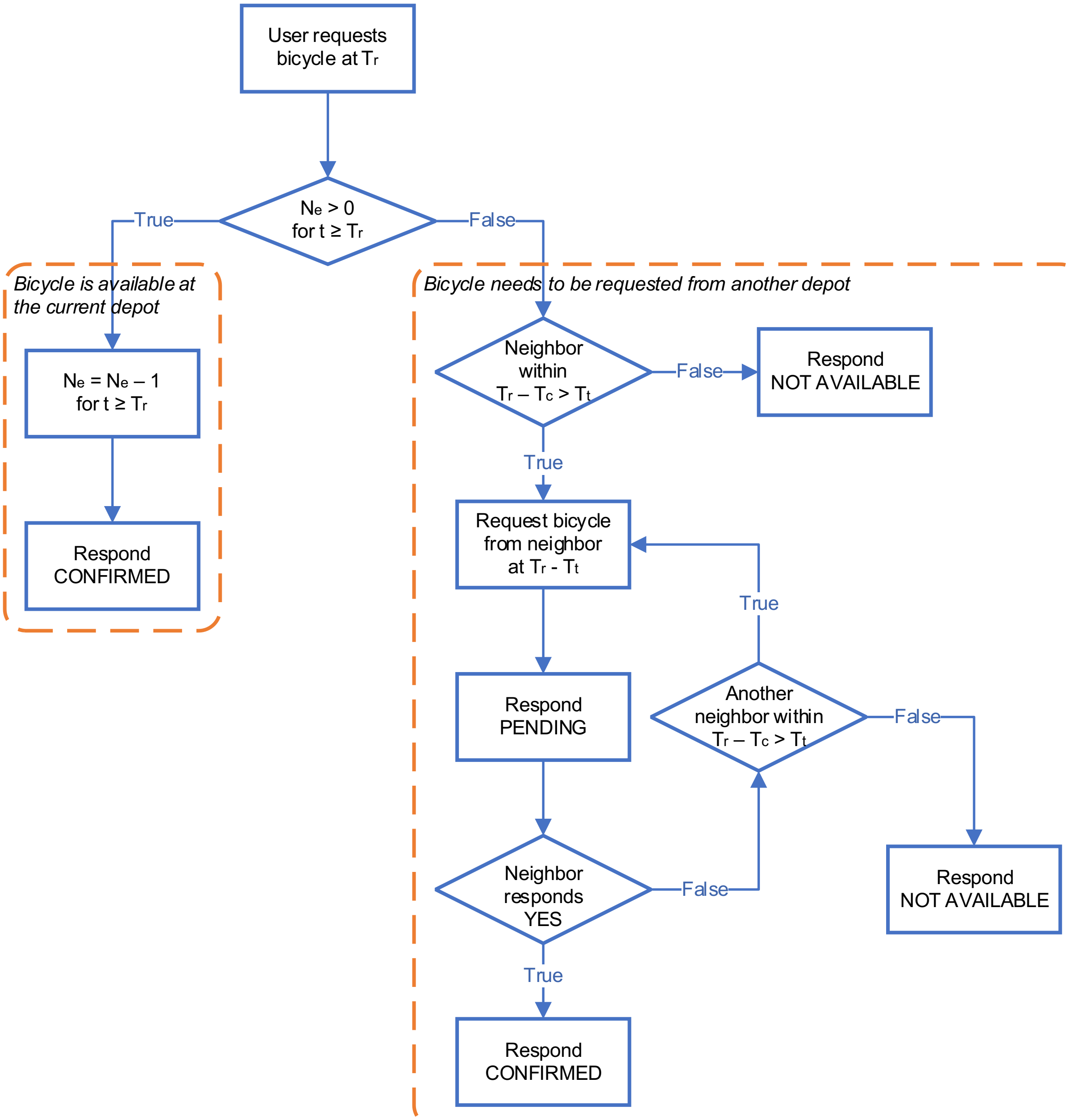


Figure 5: BIM Control Strategy for Handling Request from the User

The control strategies for handling the request from another neighbor is described the figure below:

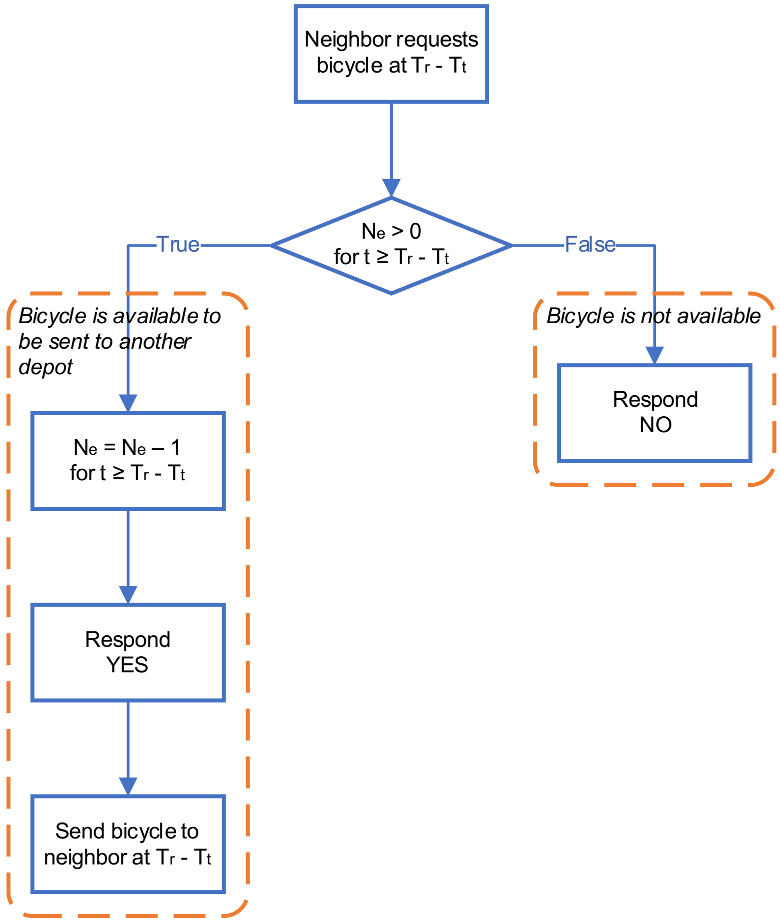


Figure 6: BIM Control Strategy for Handling Request from Another Neighbor

The control strategies for handling the user checking out the bicycle is little complicated (**Figure 7**). There are four different cases to consider: 1) The user arrives on time, and the bicycle is available, 2) The user arrives on time, but there is no bicycle available, 3) The user arrives early, and, finally, 4) The user arrives late.

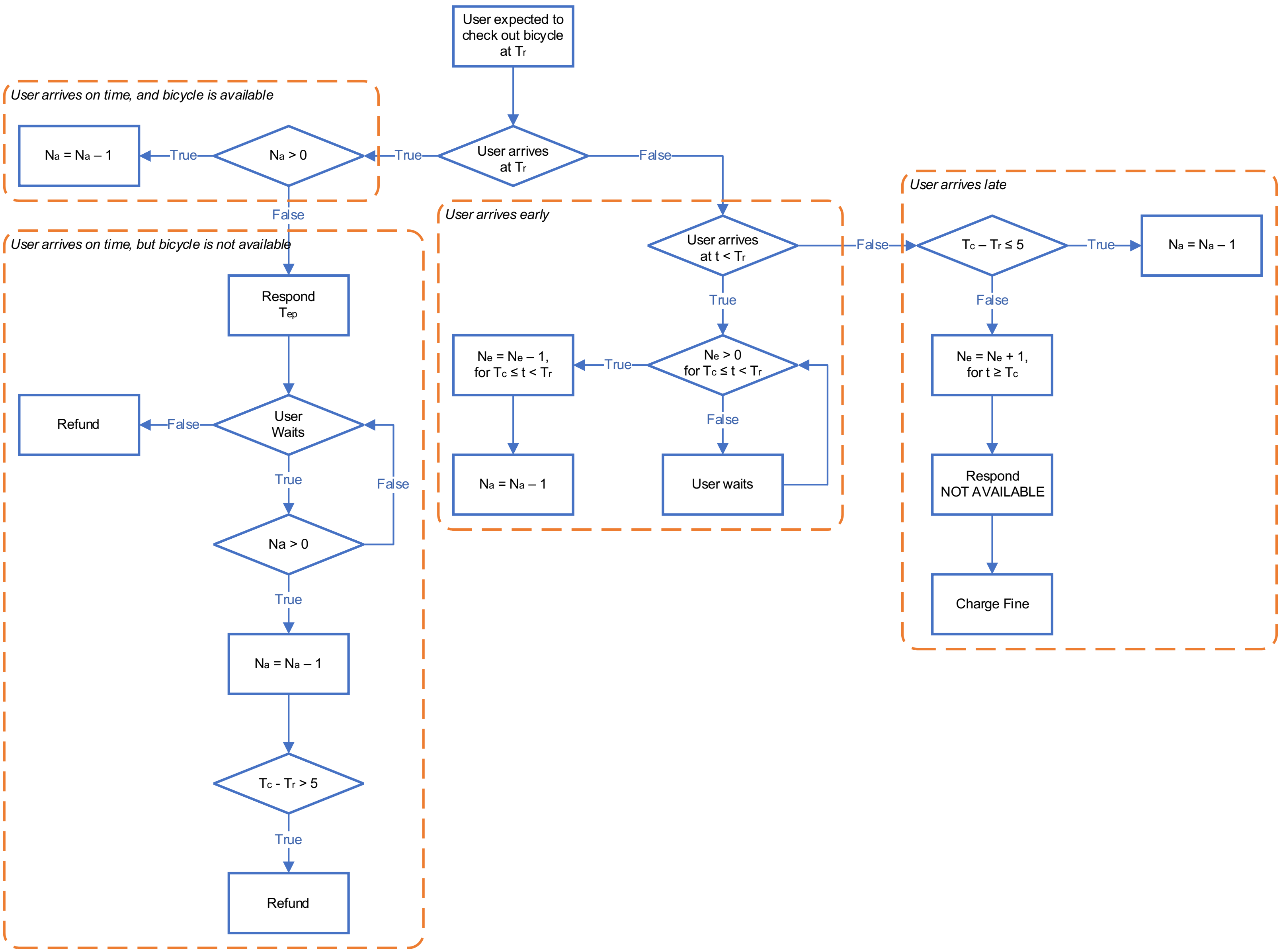


Figure 7: BIM Control Strategy for Releasing Bicycle to the User

Finally, the control strategies for handling the user returning the bicycle is described below. There are three cases: 1) The user arrives on time, 2) The user arrives early, and 3) The user arrives late (or never).

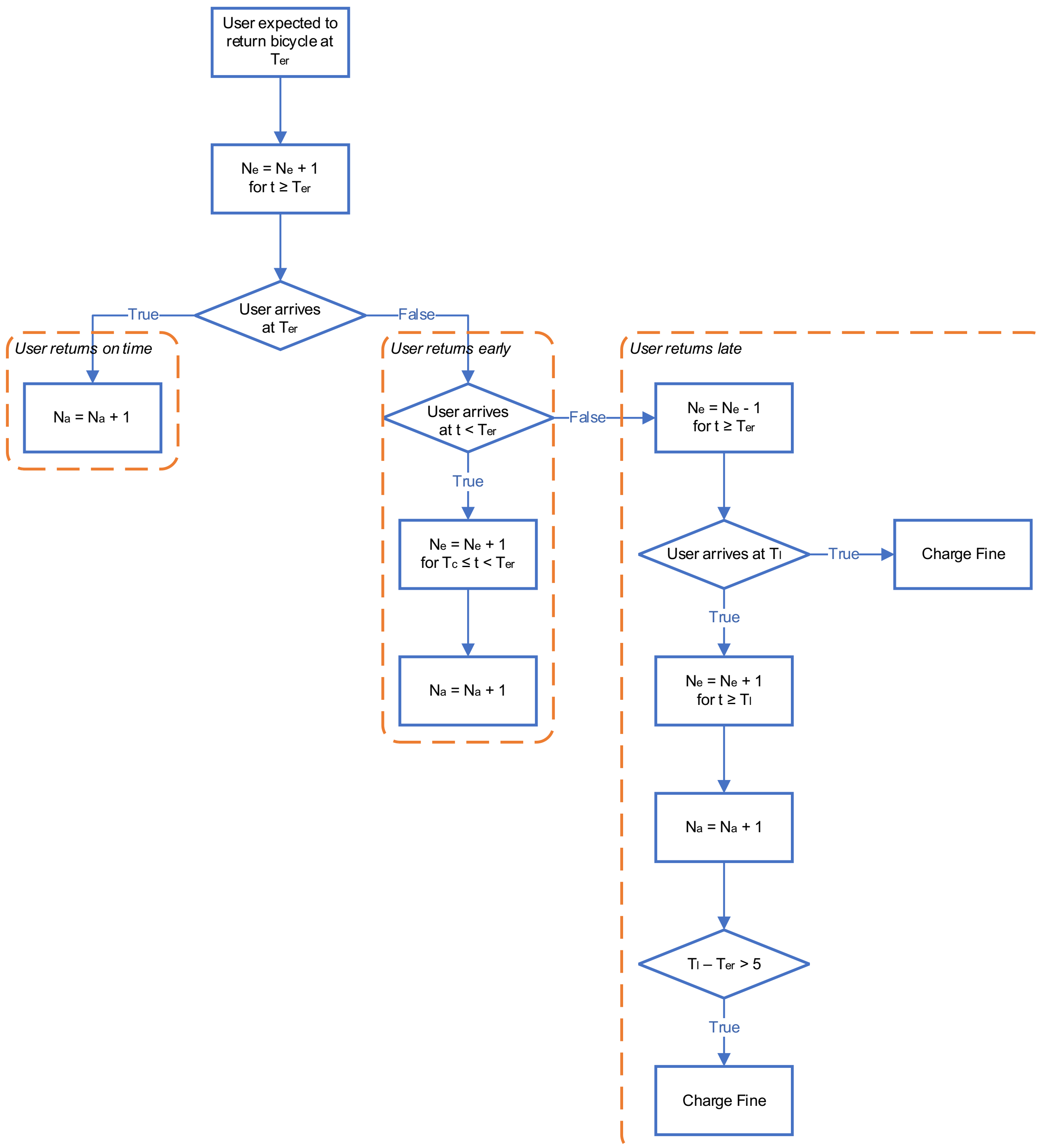


Figure 8: BIM Control Strategy for Receiving Bicycle from the User

7. Implementation

This project is implemented using C++ programming language. There are three packages in the project:

- Database (DB) package is the collection of classes that reads location, path, traffic and CatTran information from the data Comma-Separated Value (CSV) files. Classes that belong to the DB package are colored in blue in the UML class diagram below.
- Graph package is the collection of classes that are used to convert the University of Arizona campus map to the graph usable by the A* algorithm. Before the A* algorithm runs, the graph is built using the extracted data from the DB package. Classes that belong to the graph package are colored in orange in the UML class diagram below.
- AStar package includes the AStar class that runs the A* algorithm on the graph from the graph package. The AStar package is colored in yellow in the UML class diagram below.

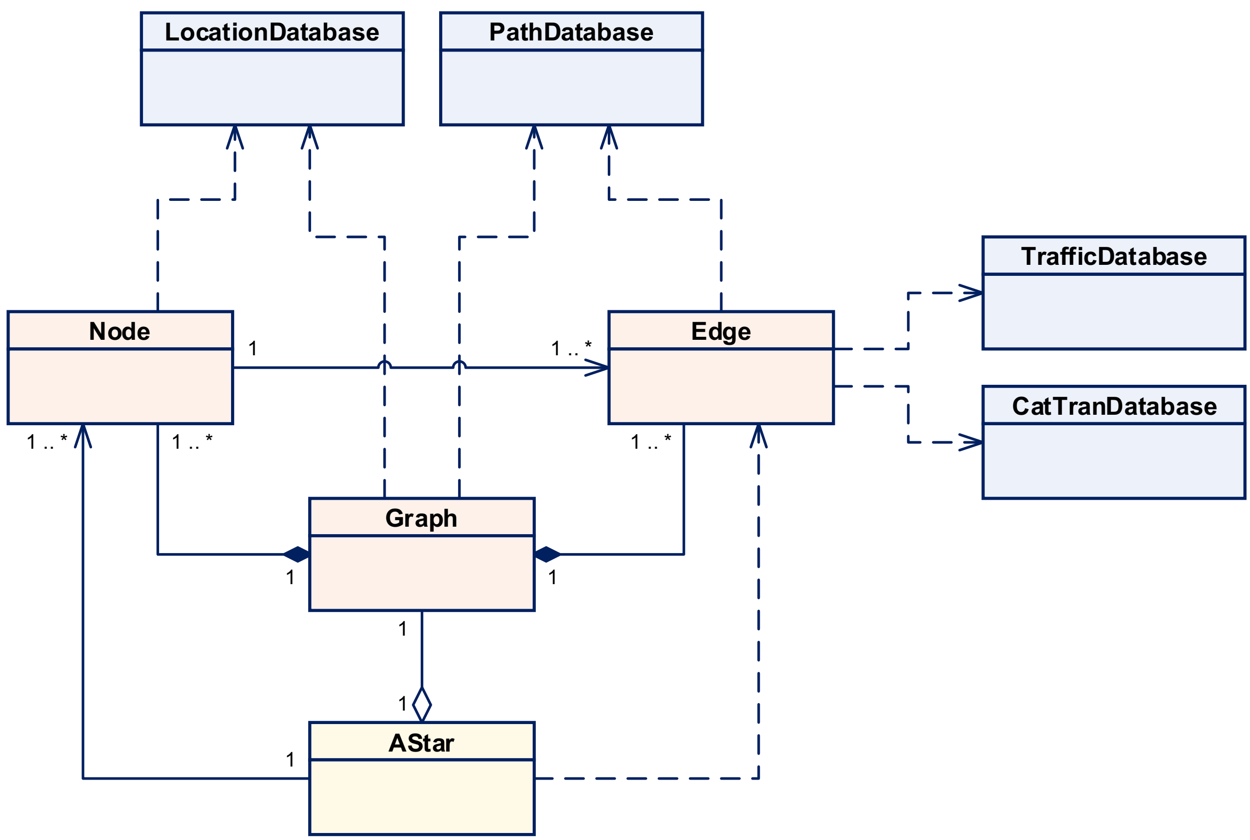


Figure 9: WIMS UML Class Diagram

More detailed UML class diagrams are included in the Appendix.

8. Results

Because bicycle is faster than CatTran (16 km/h vs. 12 km/h) in our simulation, and also because bicycle is less affected by traffic than CatTran is (traffic modal coefficient 0.7 vs. 0.3), in most scenarios, the best way to get around the campus with or without traffic is by bicycle. When we adjusted different parameters of the WIMS, we began observing more interesting results.

a. Scenario 1

In Scenario 1, the user is going from the Park Avenue Garage (Location ID: 1) to the McKale Center (Location ID: 56). In this scenario, we did not change any parameter, but compared how the traffic affects the travel time. We compared when the departure time is 8:51 am (at the peak of one of the class transition times) and when the departure time is 9:01 am (when classes just started). Even though the WIMS generated the same path and the same mode of transportation, leaving the Park Avenue Garage at 8:51 am is slower by around 36 seconds because of traffic:

Table 7: Scenario 1 Result

Departure Time: 8:51 am			
Step	**Path**	**Mode**	**Elapsed Time (Min)**
1	1 → 7	Bicycle	1.690
2	7 → 21	Bicycle	0.483
3	21 → 22	Bicycle	0.363
4	22 → 23	Bicycle	0.330
5	23 → 24	Bicycle	0.458
6	24 → 25	Bicycle	0.638
7	25 → 35	Bicycle	0.339
8	35 → 45	Bicycle	0.577
9	45 → 46	Bicycle	0.344
10	46 → 47	Bicycle	0.672
11	47 → 55	Bicycle	0.763
12	55 → 56	Walk	3.675
Total Elapsed Time (Min):			**10.332**

Departure Time: 9:01 am			
Step	**Path**	**Mode**	**Elapsed Time (Min)**
1	1 → 7	Bicycle	1.685
2	7 → 21	Bicycle	0.455
3	21 → 22	Bicycle	0.330
4	22 → 23	Bicycle	0.286
5	23 → 24	Bicycle	0.397
6	24 → 25	Bicycle	0.553
7	25 → 35	Bicycle	0.288
8	35 → 45	Bicycle	0.490
9	45 → 46	Bicycle	0.298
10	46 → 47	Bicycle	0.583
11	47 → 55	Bicycle	0.707
12	55 → 56	Walk	3.664
Total Elapsed Time (Min):			**9.736**

b. Scenario 2

In Scenario 2, the speed of CatTran is increased from 12 km/h to 20 km/h h (faster than bicycle). The user is going from the Main Library (Location ID: 52) to the AME Building (Location ID: 2). We compared two cases when the user leaves the Main Library at 7:01 am and when the user leaves the Main Library at 7:02 am. When the user leaves the Main Library at 7:01 am, it takes the user 1.20 minute to the 4th and Cherry CatTran stop, just in time for the CatTran that arrives at the stop at 7:03 am. However, when the user leaves the Main Library at 7:02 am, the user arrives at the CatTran stop around 7:03.20 and missed CatTran by about 12 seconds. The WIMS generate the following steps for the user:

Table 8: Scenario 2 Result

Departure Time: 7:01 am			
Step	**Path**	**Mode**	**Elapsed Time (Min)**
1	52 → 55	Walk	1.200
2	**55 → 47**	**CatTran**	1.524
3	**47 → 37**	**CatTran**	0.315
4	**37 → 27**	**CatTran**	0.474
5	**27 → 26**	**CatTran**	0.499
6	**26 → 25**	**CatTran**	0.532
7	**25 → 24**	**CatTran**	0.563
8	**24 → 15**	**CatTran**	0.241
9	**15 → 9**	**CatTran**	0.229
10	**9 → 4**	**CatTran**	0.504
11	**4 → 2**	**Walk**	1.310
Total Elapsed Time (Min):			**7.388**

Departure Time: 7:02 am			
Step	**Path**	**Mode**	**Elapsed Time (Min)**
1	52 → 46	Walk	2.323
2	**46 → 45**	**Bicycle**	1.454
3	**45 → 35**	**Bicycle**	0.745
4	**35 → 25**	**Bicycle**	0.438
5	**25 → 16**	**Bicycle**	0.357
6	**16 → 10**	**Bicycle**	0.335
7	**10 → 3**	**Bicycle**	1.289
8	**3 → 2**	**Walk**	1.882
Total Elapsed Time (Min):			**8.823**

c. Scenario 3

In Scenario 3, the speed of bicycle is lowered from 16 km/h to 10 km/h, and the modal traffic coefficient for bicycle is also lowered from 0.7 to 0.3 (i.e., bicycle is more affected by traffic). The user is going from the Main Library (Location ID: 52) to the ECE Building (Location ID: 6). In this scenario, we compared three different departure times: 11:25am, 12:51pm and 12:52pm. When the user departed at 11:25 am (with almost no traffic), the WIMS told the use to take bicycle whenever possible. When the user departed at 12:51 pm (when the traffic is the busiest), the WIMS told the user to walk, but follow the shortest distance path. When the user departed at 12:52pm (with the traffic little lower than departing at 12:51pm), the WIMS told the use to walk halfway and switch to bicycle for the rest of the way.

Table 9: Scenario 3 Result

Departure Time: 11:25 am			
Step	**Path**	**Mode**	**Elapsed Time (Min)**
1	52 → 46	Walk	2.432
2	46 → 45	**Bicycle**	1.664
3	45 → 35	**Bicycle**	1.092
4	**35 → 25**	**Bicycle**	0.642
5	**25 → 24**	**Bicycle**	1.238
6	**24 → 23**	**Bicycle**	0.889
7	23 → 8	**Bicycle**	0.860
8	8 → 6	Walk	1.860
Total Elapsed Time (Min):			**10.678**

Departure Time: 12:51 pm			
Step	**Path**	**Mode**	**Elapsed Time (Min)**
1	52 → 46	Walk	2.461
2	46 → 45	**Walk**	0.985
3	45 → 35	**Walk**	1.642
4	**35 → 34**	**Walk**	2.101
5	**34 → 33**	**Walk**	0.669
6	**33 → 23**	**Walk**	1.297
7	23 → 8	**Walk**	1.321
8	8 → 6	Walk	0.861
Total Elapsed Time (Min):			**11.337**

Departure Time: 12:52 pm

Step	Path	Mode	Elapsed Time (Min)
1	52 → 46	Walk	2.485
2	46 → 45	**Walk**	1.000
3	45 → 35	**Walk**	1.652
4	**35 → 25**	**Bicycle**	1.981
5	**25 → 24**	**Bicycle**	1.377
6	**24 → 23**	**Bicycle**	0.938
7	23 → 8	**Bicycle**	0.882
8	8 → 6	Walk	1.861
Total Elapsed Time (Min):			**12.178**

d. *Scenario 4*

Scenario 4 is an extreme (i.e., unrealistic) situation. We lowered the speed of bicycle to 10 km/h, lower its modal traffic coefficient to 0.3 (i.e., bicycle is slower and more affected by the traffic). We raised the speed of CatTran to 100 km/h. We also set the spatial traffic coefficient to 3.5 for rating 5 paths and to 0 for other paths (i.e., Rating 5 paths will be hyper-sensitive to traffic, while other paths will not be affected at all). In this scenario, the WIMS will prefer longer less traffic distance path with CatTran over shorter high traffic paths. In this scenario, the user is going from the Main Library to the Old Main

Table 10: Scenario 4 Result

Departure Time: 12:59 pm			
Step	**Path**	**Mode**	**Elapsed Time (Min)**
1	**52 → 46**	**Walk**	2.519
2	**46 → 45**	**Walk**	0.974
3	**45 → 44**	**Walk**	0.722
4	**44 → 43**	**Walk**	1.444
5	**43 → 42**	**Walk**	1.044
6	**42 → 41**	**Walk**	0.490
Total Elapsed Time (Min):			**7.192**

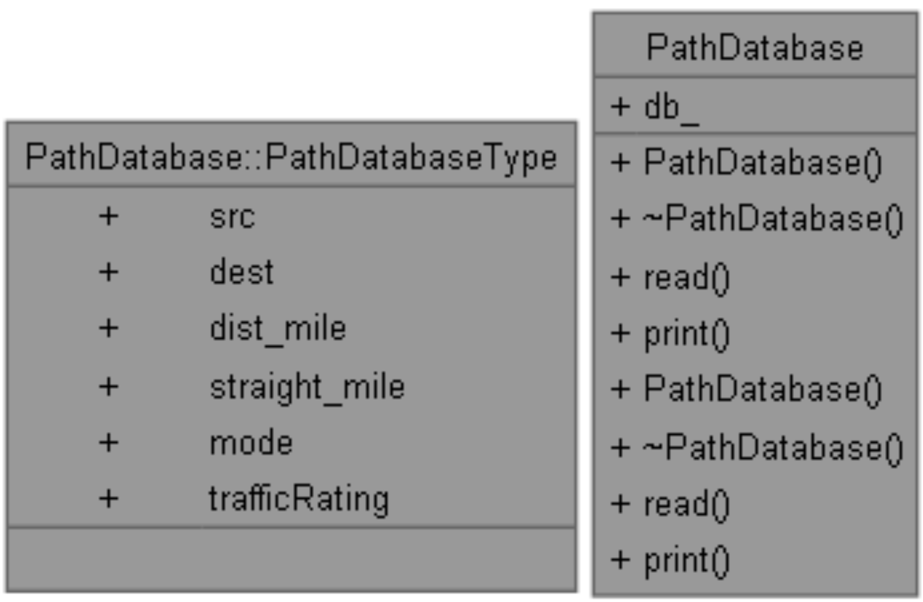
Departure Time: 13:00 pm			
Step	**Path**	**Mode**	**Elapsed Time (Min)**
1	**52 → 55**	**Walk**	1.292
2	**55 → 54**	**CatTran**	1.869
3	**54 → 57**	**CatTran**	0.142
4	**58 → 57**	**CatTran**	0.054
5	**57 → 53**	**CatTran**	0.170
6	**53 → 39**	**CatTran**	0.087
7	**39 → 40**	**CatTran**	0.109
8	**40 → 49**	**Walk**	1.218
9	**49 → 42**	**Walk**	1.511
10	**42 → 41**	**Walk**	0.490
Total Elapsed Time (Min):			**6.941**

9. Appendix

a. *In Depth UML Class Diagram*

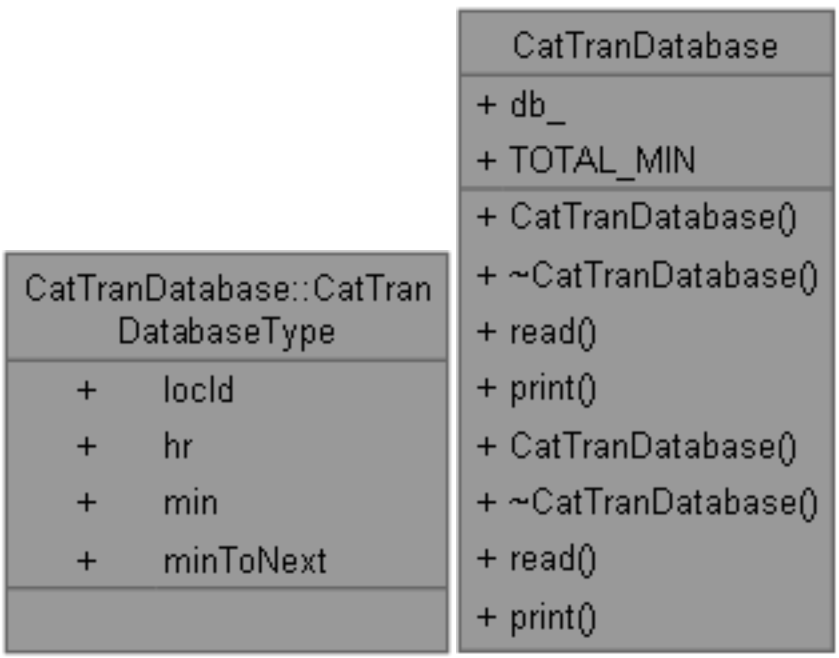
i. LocationDatabase

ii. PathDatabase

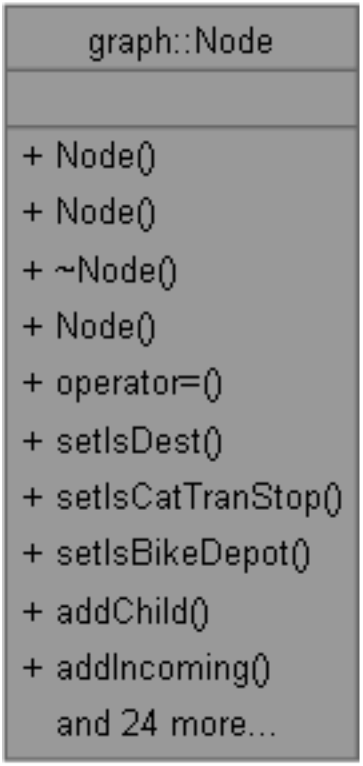


iii. TrafficDatabase

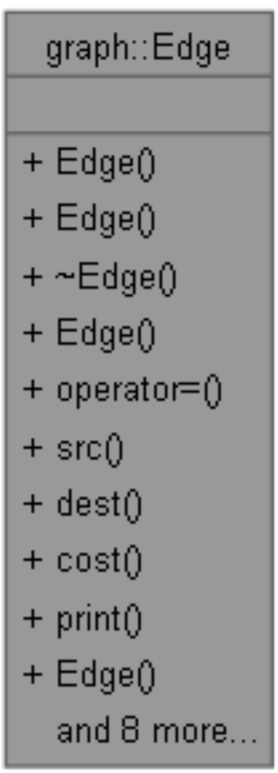
iv. CatTranDatabase



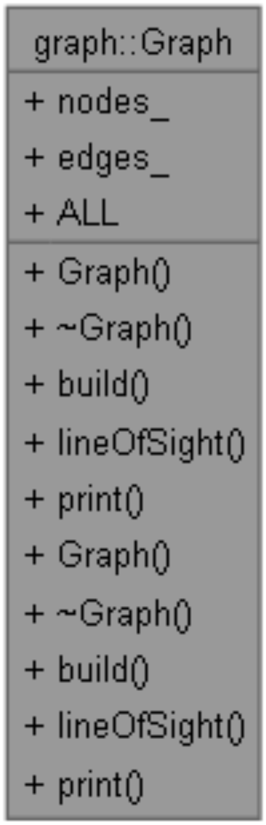
v. Node



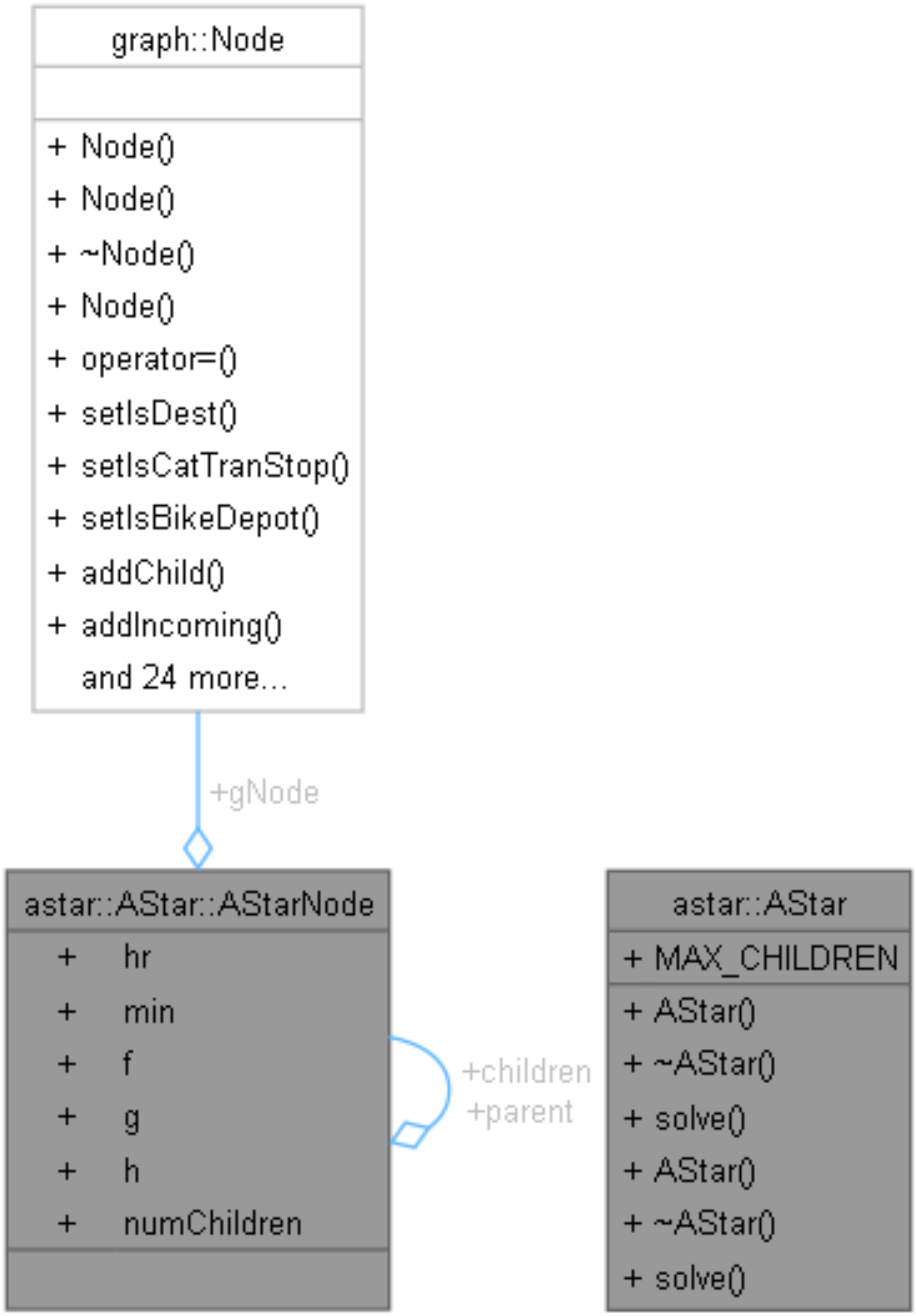
vi. Edge



vii. Graph


graph::Graph
+ nodes_
+ edges_
+ ALL
+ Graph()
+ ~Graph()
+ build()
+ lineOfSight()
+ print()
+ Graph()
+ ~Graph()
+ build()
+ lineOfSight()
+ print()

viii. Astar


graph::Node
+ Node()
+ Node()
+ ~Node()
+ Node()
+ operator=()
+ setIsDest()
+ setIsCatTranStop()
+ setIsBikeDepot()
+ addChild()
+ addIncoming()
and 24 more...
+gNode
astar::AStar::AStarNode
+ hr
+ min
+ f
+ g
+ h
+ numChildren
+children
+parent
astar::AStar
+ MAX_CHILDREN
+ AStar()
+ ~AStar()
+ solve()
+ AStar()
+ ~AStar()
+ solve()