



MAPÚA MALAYAN COLLEGES MINDANAO

JRoute: Commuting Mobile App for Jeepneys In Davao

By:

Devera, Kieffer Lance A.

Llorente, Josef Dave R.

A Research Synopsis in Partial Fulfillment of the Requirements for the IT102:
Object-Oriented Programming

Bachelor of Science in Computer Science

Submitted to:

Engr. Martzel Baste

Course Instructor

Background of the Study

Transportation is a crucial aspect of citizens' lives, determining access to education, employment, medical care, and other essential social services. Commuting has always been a popular issue for urban dwellers, especially in cities like Davao. Congestion, confusion, and inefficiency in the mobility sector have all been caused by the population increase, growth in vehicle ownership, and the lack of transportation information systems (Asian Development Bank, 2019). Commuters often find themselves unable to determine the jeepney routes, waiting times, and traffic conditions.

The number of digital tools implemented to address transport-related issues across the globe has increased. Developed countries use apps such as Google Maps, Citymapper, or Moovit to integrate real-time transport schedules and navigation (Smith and Jones, 2022). These tools guarantee commuting time as being less stressful. However, in most developing countries, particularly in the urban centers such as Davao, commuters are still resorting to the traditional means, such as inquiring passersby, memorizing routes, or even relying on word of mouth. This is a hindrance to first-time travelers, students, and even the local workers, who also need an effective and reliable mode of transportation.

These problems are met at the local level. In our case, as commuters in Davao, we experienced confusion when using jeepney routes, particularly when changing lanes to arrive at the destination. Indicatively, when we tried to join the right jeepney to the university, we had to repeat ourselves to drivers or passengers at the road. Such unstructured information wastes time and may be overwhelming to tourists or first-time visitors in the city. These obstacles emphasize the significance of a digital platform that offers real-time and direct commuting data, which accommodates the system of jeepneys of Davao.

Therefore, the proponents propose to implement JRoute, the mobile application that would target commuters in the city of Davao. The app will provide the introduction of a complex system of navigation, mapping of all jeepneys, and real-time information to make transportation efficient, accessible, and convenient.

Sustainable Development Goals (SDGs)

The proposed **JRoute mobile application** supports several United Nations Sustainable Development Goals by addressing issues of mobility, accessibility, and sustainability in Davao City. Primarily, it aligns with **SDG 11: Sustainable Cities and Communities**, as it promotes efficient and inclusive urban transportation, making commuting safer and more convenient for both residents and visitors. Lastly, by reducing wasted commuting time and improving access to economic opportunities, JRoute supports **SDG 8: Decent Work and Economic Growth**.

Problem Statement

Commuters in Davao City are also faced with numerous challenges because informational transport systems are not available. These include a lack of clarity in determining right and wrong jeepney routes, a lack of real-time updates on traffic, as well as delays and limited digital resources that cater specifically to local transport systems. This often results in wasted time, unnecessary cost, and stress to commuters.

Without a major commuter-specific platform, the city is in a dilemma to address the problem of accessibility and efficiency within the public transportation sector to its fullest potential. For newcomers and tourists, navigating the jeepney system is much more complicated with the lack of signage and route clarity. Therefore, a need for a mobile solution arises that combines route mapping, real-time tracking, and simplified navigation to improve the commuting experience.

Project Goals and Objectives

- **To develop a user-friendly commuter application** tailored to Davao's jeepney system, allowing easy navigation for both locals and visitors.
- **To provide real-time route guidance and estimated travel times**, helping commuters plan their trips more effectively.
- **To integrate GPS-based navigation and mapping** for jeepney routes to reduce confusion and errors in commuting
- **To encourage public transportation usage** as a cost-effective and eco-friendly alternative to private vehicles.
- **To support sustainable urban mobility initiatives** of Davao City by promoting innovation and digital infrastructure in the transport sector.

Conceptual Framework

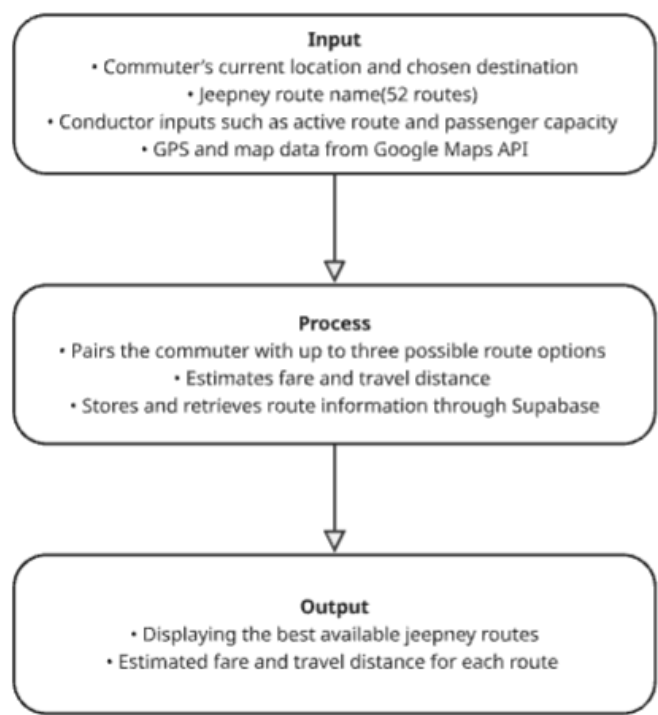


Figure 1

Target Market

The JRoute mobile application target market mainly consists of people who use jeepneys regularly in Davao City. This has a broad demographic base that includes students, employees, tourists, and residents who use public transport to meet their daily mobility requirements.

Primary Users

- Students – From universities and colleges across Davao who commute daily and often struggle with unfamiliar jeepney routes.
- Employees – Particularly those working in urban districts such as downtown, Bajada, or Matina, who aim to save time and plan more efficient routes to and from work.
- Tourists and First-Time Visitors – Individuals visiting Davao who are unfamiliar with the city’s jeepney system and routes.

Secondary Users

- Jeepney Drivers and Operators – who may use the app to monitor passenger flow, improve coordination, and identify high-demand routes.
- Local Government Units (LGUs) – particularly the City Transport and Traffic Management Office (CTTMO), which can utilize aggregated data to plan transport improvements.

The JRoute app thus caters to both commuters seeking convenience and stakeholders promoting urban mobility efficiency, contributing to a more connected and commuter-friendly Davao City.

Similar Applications

	Google Maps	Moovit	Citymapper	Sakay.ph	JTransit	Jroute
Real-Time Tracking	✔	✔	✔	✔	✔	✔
Jeepney Route Mapping	✘	✘	✘	✔	✔	✔
Estimated Travel Time	✔	✔	✔	✔	✔	✘
Offline Navigation	✔	✘	✔	✘	✔	✘
Fare Calculation	✔	✔	✔	✔	✘	✔
Multiple Route Options	✔	✔	✔	✔	✘	✔

Figure 2

Scope and Limitations

The JRoute application is a jeepney routing application developed by the city of Davao, which aims to popularize cultural and sustainable ways of transportation (instead of taxis and cars). The project mainly aims at building a mobile application that will have two primary user roles, namely Commuter and Conductor. The Commuter interface assists people with the closest available jeepney routes, the estimated cost and distance to travel, and a maximum of three different route patterns that they could use to get to their destination. At the same time, the Conductor interface enables jeepney drivers to display their active routes and their passenger capacity. The front-end and cross-platform development is based on Flutter, a cloud database on Supabase, and mapping and navigation with the help of Google Maps APIs. They will also be confined to the 52 established jeepney routes in Davao City, which include key locations such as Bajada, Matina, Poblacion, and Toril.

The existing system has a number of limitations. The accuracy of the application relies on the strength of the GPS signal, the connection of the mobile data, and manually placed route data, which is not always the case in the real world through one-way roads or obstacles. The nearest-route detection algorithm is based on geometric approximation and cannot recognize wrong routes in compactly dense regions. The app is also not yet optimized to work with iOS and is currently only available to Android users. The system lacks real-time tracking because jeepneys in the Philippines are not yet equipped with IoT devices for location monitoring. Additional improvements can be made in the future through the implementation of GeoJSON route mapping, IoT-enabled real-time tracking, more accurate routing algorithms, the addition of other transport modes (e.g., tricycles and buses), and the mapping of the UI/UX to make the navigation easier. In spite of these

limitations, JRoute provides a solid ground on how urban mobility can be transformed with the help of digital innovation and data-driven management of public transport.

Project Definition (Tools, Frameworks, Libraries)

Platform & Language

- Frontend & shared logic: Dart / Flutter (cross-platform UI framework).
- Backend / Database: Supabase (Postgres + Auth + storage).
- Mapping & geocoding: Google Maps Platform (Maps SDK / Places API / Geocoding).
- Version control/repo: GitHub.

Key Libraries / Packages

- flutter (UI & navigation)
- google_maps_flutter (map widget)
- google_place / google_maps_webservice (places & geocoding)
- supabase_flutter (auth, realtime, database client)
- geolocator (device GPS/location)
- flutter_riverpod or provider (state management — choose one)
- flutter_map (optional alternative mapping)
- json_serializable / freezed (data models, immutability & (de)serialization)
- dio or http (network requests)
- mapbox_gl (optional alternative)

Development Tools

Android Studio / VS Code, Flutter CLI, Postman for API/DB testing, QGIS (recommended for future GeoJSON route prep).

OOP and GUI Implementation Details

OOP Implementation:

JRoute uses **Object-Oriented Programming principles** for structured and maintainable code. Key classes include:

- **Commuter** – Handles input of origin and destination, displays nearby routes, and shows fare/distance estimates.
- **Conductor** – Inputs current route, maximum passengers, and shares location for pairing with commuters.
- **Route** – Represents each jeepney route with attributes such as routeID, routeName, stops, and distance.
- **MapService** – Manages GPS integration, route calculations, and nearest-route suggestions using Google Maps API.
- **DatabaseService** – Interfaces with Supabase to store route and ride data.

Key OOP concepts

- **Encapsulation**: Route and ride data are private with public getters/setters.

- **Inheritance:** Commuter and Conductor share base behaviors for interacting with routes.
- **Polymorphism:** Methods like displayRoute() are customized for each user type.
- **Abstraction:** Map and database operations are encapsulated into service classes for reusability.

GUI Implementation:

The **Flutter UI** provides:

- **Home Screen** – Allows selection of Commuter or Conductor role.
- **Commuter Page** – Origin/destination input, map view of nearby routes, fare and distance info.
- **Conductor Page** – Input active route and max passengers, view route info.
- **Navigation Controls** – Quick access to route suggestions and map interaction.

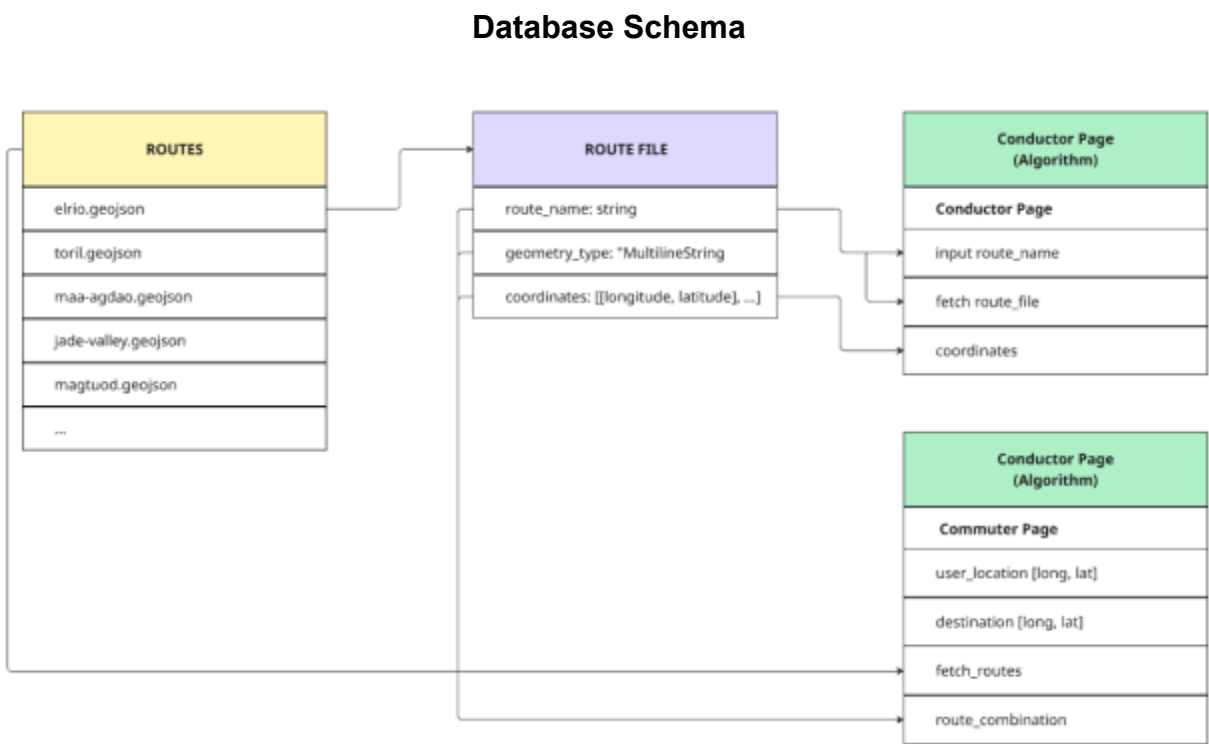


Figure 3

CRUD Functionalities

The fundamental processes that control the system's data flow are represented by the CRUD functions (Create, Read, Update, Delete). Through the use of local caching and effective rendering techniques to maximize performance, these actions in our application guarantee the seamless handling of map and route data.

- **Create** – This function primarily handles map rendering, which involves loading and displaying route data on the map interface. Instead of traditional data entry, the system displays visual routes dynamically, allowing users to see available jeepney routes in Davao City.
- **Read** - The Read operation retrieves data from a cloud database and caches it locally to reduce redundant data retrieval. This allows for faster access to route information and enhances performance, particularly when examining many routes in a short period of time.
- **Update** - The update procedure involves real-time map rendering and route change.

When users interact with or change the map, the displayed information is dynamically updated to reflect the most recent cached data.

- **Delete** - This function trims or segments GeoJSON data, deleting old or unneeded features. This guarantees that the system has only relevant and up-to-date information, hence ensuring efficiency and data correctness.

Prototyping Screenshots

.NET Maui prototype:

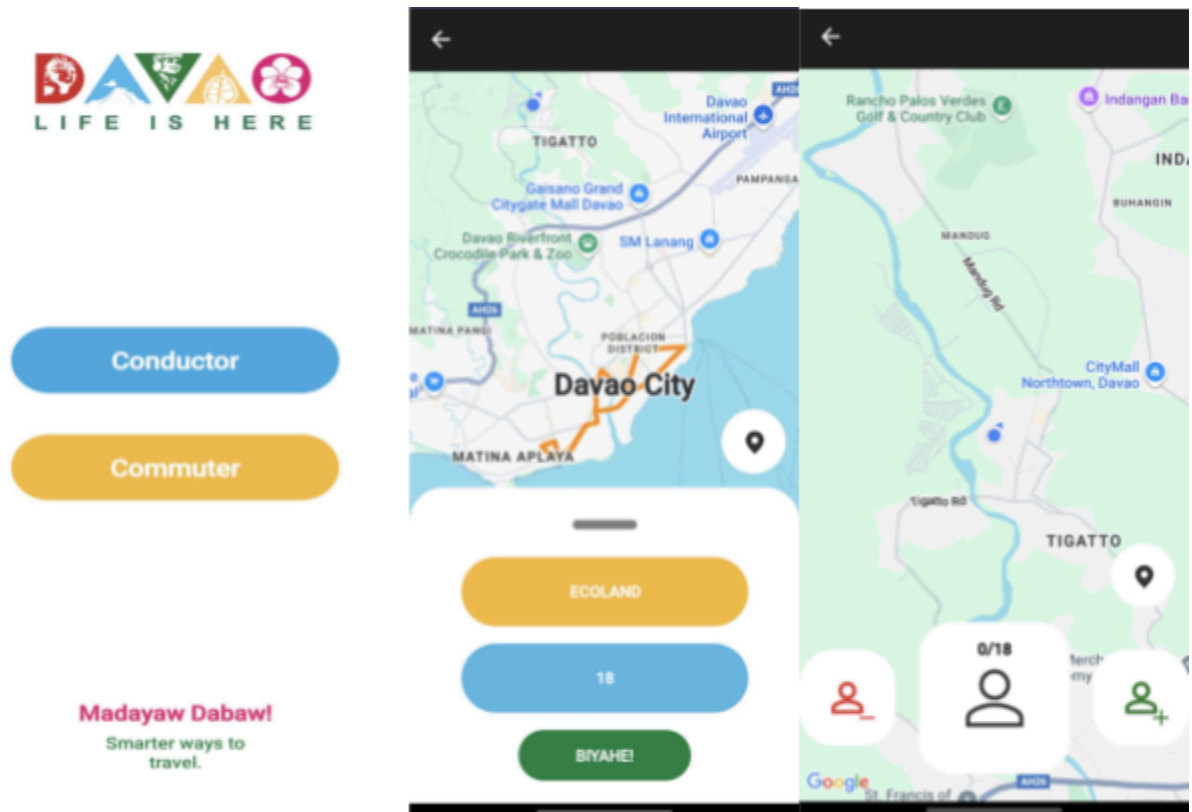


Figure 4

Sample Input/Output

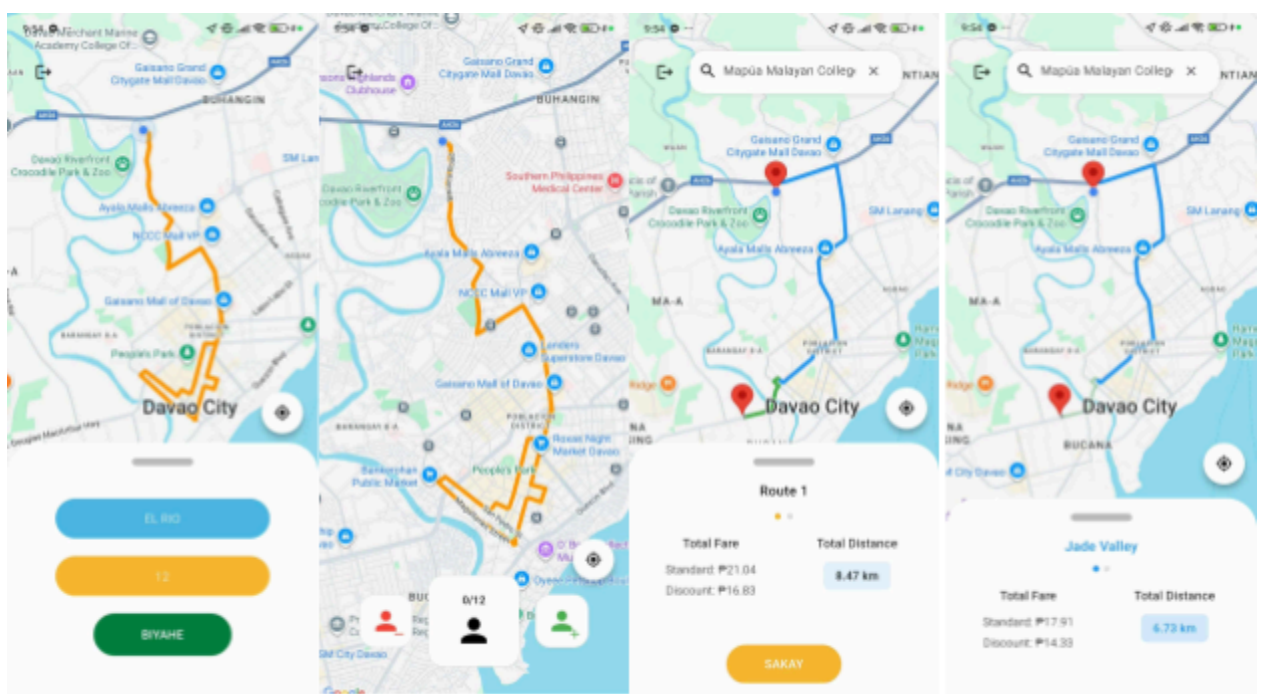


Figure 5

Review of Related Literature

Public transportation is a critical component of urban mobility, and digital applications have increasingly played a key role in improving efficiency and accessibility. Mobile

applications provide real-time information on public transport, enabling users to plan journeys more efficiently and respond to changes in service conditions (Santos & Nikolaev, 2021). Commuters also benefit from improved accessibility and reduced waiting times when public transport is integrated with digital journey-planning tools (Santos & Nikolaev, 2021). In terms of design and usability, all interviewees in a Citymapper case study used at least one app to help plan their journey (Google Maps, Apple Maps, CityMapper, or Moovit), though limitations were mentioned for every single one (Viola, 2023).

In the Philippine context, commuting students face traffic congestion, long travel times, safety issues, and factors that affect stress levels and academic engagement (Morales, Arellano, & Giron, 2024), demonstrating the need for efficient, real-time digital solutions. Applications designed for public transport also support sustainable development by improving efficiency, convenience, and sustainability, while reducing traffic congestion, air pollution, and greenhouse gas emissions (Shukirillaeva & Tokhirov, 2023). Despite these benefits, challenges persist in developing cities where many still lack fit-for-purpose regulation for app-based mobility services (Ireland, Combe, & Montasser, 2024). These studies collectively highlight the need for localized digital tools that enhance commuting, reduce stress, and promote sustainable urban transport, supporting the development of solutions like *JRoute* for Davao City.

Results and Discussion

Test Cases	Origin (Control Variable)	Destination	Suggested Route	Correct Routes Found	Accuracy (%)
1	El Rio	4U Grocery (Bacaca Road)	1	1	100%
2	El Rio	University of Southeastern Philippines (Obrero)	2	2	100%
3	El Rio	SM Lanang (Lanang)	2	2	100%
4	El Rio	Mapua Malayan Colleges Mindanao (Gen. Douglas MacArthur Hwy)	2	2	100%
5	El Rio	Crocodile Park (AH26)	1 (Fallback Route)	1	100%
6	El Rio	Tito Nat's Super Balbacua Maa (Ma-a Road)	1 (Fallback Route)	1	100%
7	El Rio	Lyceum of the Philippines Davao (C.P. Garcia Highway)	1	1	100%

Figure 6

The table's results show that the algorithm is quite effective at identifying correct and optimal routes from the origin site, El Rio. Across all seven test cases, the algorithm maintained a 100% accuracy rate, indicating that the suggested routes consistently matched the correct routes discovered. This consistency indicates the trustworthiness of the routing algorithm as well as the efficiency of the program's data-fetching and caching methods.

However, during testing, certain destinations, such as Crocodile Park (AH26) and Tito Nat's Super Balbacua Maa (Ma-a Road), were discovered to be accessible via existing routes, but the algorithm failed to display these routes correctly. This problem stems from the GeoJSON data being saved in a multiline format, which complicates how the system parses and shows the routes. These issues can be resolved by parsing the GeoJSON data into a more appropriate data format, guaranteeing that all reachable destinations are correctly presented in subsequent cycles. Despite this slight constraint, the algorithm continues to provide accurate and efficient jeepney route suggestions in Davao City, with room for improvement.

Conclusion

JRoute mobile app is a solution that is able to solve the problems of commuters in the jeepney network of Davao City. The testing of the prototypes and the feedback provided by the users prove that the app delivers precise route recommendations, real-time travel details, and user-friendly navigation, contributing to the minimization of confusion, the time spent by the commuters, and the stress levels of the conductors. Its easy-to-use interface enables those local and first-time travelers to plan their trips effectively, and the incorporation of GPS-based route maps makes it more accessible in the whole city. Although there was route inaccuracy and device incompatibility, the final output is that JRoute is a feasible and realistic digital tool to enhance urban mobility. These could serve as evidence of the possibilities of localized transport applications to improve the commuter experience, encourage the use of public transport, and to allow increasing sustainable urban mobility in Davao City.

Recommendations

- **Route Data Accuracy:** Integrate GeoJSON mapping and QGIS to refine route geometries and increase nearest-route detection accuracy.
- **Real-Time Updates:** Implement IoT or live GPS tracking for conductors to provide more precise real-time information.
- **UI/UX Improvements:** Optimize the interface for clarity, especially for first-time users or tourists unfamiliar with local jeepney routes.
- **Device Testing:** Expand testing to iOS devices and additional Android models to ensure performance consistency.

- Future Features: Include multiple vehicle types (tricycles, buses) and advanced route recommendations considering real-world obstacles and traffic conditions.

References

- CEDTyClea. (2019, September 26). Metro Manila most congested developing Asian city — ADB. *BusinessWorld* Online. <https://www.bworldonline.com/editors-picks/2019/09/26/255757/metro-manila-most-congested-developing-asian-city-adb/>
- Ireland, M., Combe, C., & Montasser, M. (2024). Regulating app-based mobility: Case studies from Bangkok, Manila, and Phnom Penh. In E. Windisch & C. Larmanou (Eds.), *International Transport Forum Policy Papers* (Case-Specific Policy Analysis No. 139). OECD Publishing. https://www.itf-oecd.org/sites/default/files/docs/regulating-app-mobility-bangkok-manila-phnom-penh_0.pdf
- Morales, N. S., Arellano, J. E., & Giron, G. L. (2024). Challenges of commuting via public transportation: An analysis of students traveling to National University-Manila from the South. https://www.researchgate.net/publication/386898211_Challenges_of_Commuting_via_Public_Transportation_An_Analysis_of_Students_Traveling_to_National_University-Manila_from_the_South
- Santos, G., & Nikolaev, N. (2021). Mobility as a service and public transport: A rapid literature review and the case of Moovit. *Sustainability*, 13(7), 3666. <https://doi.org/10.3390/su13073666>
- Shukirillaeva, M., & Tokhirov, E. (2023). The importance of applications for public transport. *Universum: Technical Sciences*, 110. https://www.researchgate.net/publication/372097520_THE_IMPORTANCE_OF_APPLICATIONS_FOR_PUBLIC_TRANSPORT
- Viola. (2023). Improving public transport with design thinking: A Citymapper case study. *Medium*. <https://medium.com/design-bootcamp/improving-public-transport-with-design-thinking-a-citymapper-case-study-70e849bf7e3f>

Appendices

Figures:

Figure 1 - Conceptual Framework

Figure 2 - Similar Applications

Figure 3 - Database Schema

Figure 4 - Prototype Screenshots

Figure 5 - Sample Input/Output

Figure 6- Results

Source Code

Main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_dotenv/flutter_dotenv.dart';
3 import 'login.dart';
4 import 'package:supabase_flutter/supabase_flutter.dart';
5
6
7 Future<void> main() async {
8   WidgetsFlutterBinding.ensureInitialized();
9   await dotenv.load(fileName: '.env');
10  await Supabase.initialize(
11    url: dotenv.env['SUPABASE_URL']!,
12    anonKey: dotenv.env['SUPABASE_ANON_KEY']!,
13  );
14
15  runApp(const MyApp());
16 }
17
18 class MyApp extends StatelessWidget {
19   const MyApp({super.key});
20
21   @override
22   Widget build(BuildContext context) {
23     return MaterialApp(
24       title: 'J Route',
25       debugShowCheckedModeBanner: false,
26       theme: ThemeData(primarySwatch: Colors.blue),
27       home: const LoginPage(),
28     );
29   }
30 }
```

Login.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:provider/provider.dart';
3 import 'package:conductor.dart';
4 import 'services/routing_service.dart';
5 import 'package:login/login.dart';
6 import 'services/button_transition.dart'; // added import
7
8 class LoginPage extends StatelessWidget {
9   const LoginPage({Key? key}) : super(key: key);
10
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       backgroundColor: Colors.white,
15       body: SafeArea(
16         // use a Stack so we can place a gradient "bliss" at the very top behind content
17         child: Stack(
18           children: [
19             // Gradient bliss positioned at the very top
20             Positioned(
21               top: -40,
22               left: -40,
23               child: Container(
24                 width: 320,
25                 height: 120,
26                 decoration: BoxDecoration(
27                   gradient: RadialGradient(
28                     colors: [
29                       const Color.fromRGBO(255, 255, 255, 1),
30                       const Color.fromRGBO(255, 255, 255, 1),
31                       const Color.fromRGBO(255, 255, 255, 1),
32                     ],
33                     stops: const [0.0, 0.5, 1.0],
34                     center: const Alignment(-0.2, -0.2),
35                     radius: 0.5,
36                   ),
37                   borderRadius: BorderRadius.circular(100),
38                 ),
39               // slight blue-line effect using boxshadows
40               boxShadows: [
41                 BoxShadow(
42                   color: Colors.blue.withOpacity(0.05),
43                   blurRadius: 40,
44                   spreadRadius: 10,
45                 ),
46               ],
47             ),
48             // multiple gradient blisks positioned at the very top
49             Positioned(
50               top: -140,
51               left: -140,
52               child: Container(
53                 width: 360,
54                 height: 180,
55                 decoration: BoxDecoration(
56                   gradient: RadialGradient(
57                     colors: [
58                       const Color.fromRGBO(255, 255, 255, 1),
59                       const Color.fromRGBO(255, 255, 255, 1),
60                       Colors.white,
61                     ],
62                     stops: const [0.0, 0.5, 1.0],
63                     center: const Alignment(-0.5, -0.5),
64                     radius: 0.5,
65                   ),
66                   borderRadius: BorderRadius.circular(100),
67                 ),
68               boxShadows: [
69                 BoxShadow(
70                   color: Colors.blue.withOpacity(0.05),
71                   blurRadius: 40,
72                   spreadRadius: 10,
73                 ),
74               ],
75             ),
76             // multiple gradient blisks positioned at the very top
77             Positioned(
78               top: -140,
79               left: -140,
80               child: Container(
81                 width: 360,
82                 height: 180,
83                 decoration: BoxDecoration(
84                   gradient: RadialGradient(
85                     colors: [
86                       const Color.fromRGBO(255, 255, 255, 1),
87                       const Color.fromRGBO(255, 255, 255, 1),
88                       Colors.white,
89                     ],
90                     stops: const [0.0, 0.5, 1.0],
91                     center: const Alignment(-0.5, -0.5),
92                     radius: 0.5,
93                   ),
94                   borderRadius: BorderRadius.circular(100),
95                 ),
96               boxShadows: [
97                 BoxShadow(
98                   color: Colors.blue.withOpacity(0.05),
99                   blurRadius: 40,
100                  spreadRadius: 10,
101                ),
102              ],
103            ),
104          ],
105        ),
106      ),
107    );
108  }
109}
110
111 // CONDUCTOR button: capture its position with Builder
112 Builder(builder: (BuildContext) {
113   return SizedBox(
114     width: 300,
115     child: ElevatedButton(
116       style: ElevatedButton.styleFrom(
117         backgroundColor: const Color.fromRGBO(255, 255, 255, 1),
118         padding: const EdgeInsets.symmetric(vertical: 20),
119       ),
120       onPressed: () {
121         // get the button's global rect
122         final RenderBox box = context.findRenderObject() as
123         RenderBox;
124         final Rect rect = box.localToGlobal(Offset.zero) & box.size;
125       },
126     ),
127   );
128}
129
130 // car_loading.gif centered in the middle and not overlapped by blisks or buttons
131 // positioned with top/button constraints to avoid the top blisks and bottom buttons
132 Positioned(
133   top: 120,
134   bottom: 100,
135   left: 0,
136   right: 0,
137   child: Center(
138     child: Image.asset(
139       'assets/Red Car.jpg',
140       height: 100,
141       fit: BoxFit.contain,
142       repeat: true,
143     ),
144   ),
145 )
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```

1 import 'package:flutter/material.dart';
2 import 'package:geolocator/geolocator.dart';
3 import 'package:google_maps_flutter/google_maps_flutter.dart';
4 import 'components/button_model.dart';
5 import 'components/input_button.dart';
6 import 'components/passenger.dart';
7 import 'components/vehicle.dart';
8 import 'services/conductor_map_render.dart';
9 import 'utils.dart'; // add this import near the top
10
11 class ConductorPage extends StatefulWidget {
12   const ConductorPage({Key? key}) : super(key: key);
13
14   @override
15   _ConductorPageState createState() => _ConductorPageState();
16 }
17
18 class _ConductorPageState extends State<ConductorPage> {
19   Map<LatLng, bool> _mapController;
20   final _testMapController = TestMapController();
21   final _testMapController = _testMapController();
22   final _testMapController = _testMapController();
23   final _testMapController = _testMapController();
24   final _testMapController = _testMapController();
25   final _testMapController = _testMapController();
26
27   // start expanded on load
28   bool _isExpanded = true;
29   String _autocompleteText = '';
30   bool _isExpandedAuto = false;
31   bool _isExpandedMap = false;
32   bool _isExpandedRoute = false;
33   bool _isExpandedPassenger = false; // track if map button was pressed
34   int _autocompleteCount = 0;
35   bool _autocompleteEnabled = false;
36
37   static const List<LatLng> _routes = [
38     'MADRID ATLANTA',
39     'MADRID',
40     'MADRID MADRID',
41     'MADRID VIA MADRID',
42     'MADRID VIA MADRID',
43     'MADRID VIA MADRID',
44     'MADRID VIA MADRID',
45     'MADRID VIA MADRID',
46     'MADRID VIA MADRID',
47     'MADRID VIA MADRID',
48     'MADRID VIA MADRID',
49     'MADRID VIA MADRID',
50     'MADRID VIA MADRID',
51     'MADRID VIA MADRID',
52     'MADRID VIA MADRID',
53     'MADRID VIA MADRID',
54     'MADRID VIA MADRID',
55     'MADRID VIA MADRID',
56     'MADRID VIA MADRID',
57     'MADRID VIA MADRID',
58     'MADRID VIA MADRID',
59     'MADRID VIA MADRID',
60     'MADRID VIA MADRID',
61     'MADRID VIA MADRID',
62     'MADRID VIA MADRID',
63     'MADRID VIA MADRID',
64     'MADRID VIA MADRID',
65     'MADRID VIA MADRID',
66     'MADRID VIA MADRID',
67     'MADRID VIA MADRID',
68     'MADRID VIA MADRID',
69     'MADRID VIA MADRID',
70     'MADRID VIA MADRID',
71     'MADRID VIA MADRID',
72     'MADRID VIA MADRID',
73     'MADRID VIA MADRID',
74     'MADRID VIA MADRID',
75     'MADRID VIA MADRID',
76     'MADRID VIA MADRID',
77     'MADRID VIA MADRID',
78     'MADRID VIA MADRID',
79     'MADRID VIA MADRID',
80     'MADRID VIA MADRID',
81     'MADRID VIA MADRID',
82     'MADRID VIA MADRID',
83     'MADRID VIA MADRID',
84     'MADRID VIA MADRID',
85     'MADRID VIA MADRID',
86     'MADRID VIA MADRID',
87     'MADRID VIA MADRID',
88     'MADRID VIA MADRID',
89     'MADRID VIA MADRID',
90     'MADRID VIA MADRID',
91     'MADRID VIA MADRID',
92     'MADRID VIA MADRID',
93     'MADRID VIA MADRID',
94     'MADRID VIA MADRID',
95     'MADRID VIA MADRID',
96     'MADRID VIA MADRID',
97     'MADRID VIA MADRID',
98     'MADRID VIA MADRID',
99     'MADRID VIA MADRID',
100    'MADRID VIA MADRID',
101    'MADRID VIA MADRID',
102    'MADRID VIA MADRID',
103    'MADRID VIA MADRID',
104    'MADRID VIA MADRID',
105    'MADRID VIA MADRID',
106    'MADRID VIA MADRID',
107    'MADRID VIA MADRID',
108    'MADRID VIA MADRID',
109    'MADRID VIA MADRID',
110    'MADRID VIA MADRID',
111    'MADRID VIA MADRID',
112    'MADRID VIA MADRID',
113    'MADRID VIA MADRID',
114    'MADRID VIA MADRID',
115    'MADRID VIA MADRID',
116    'MADRID VIA MADRID',
117    'MADRID VIA MADRID',
118    'MADRID VIA MADRID',
119    'MADRID VIA MADRID',
120    'MADRID VIA MADRID',
121    'MADRID VIA MADRID',
122    'MADRID VIA MADRID',
123    'MADRID VIA MADRID',
124    'MADRID VIA MADRID',
125    'MADRID VIA MADRID',
126    'MADRID VIA MADRID',
127    'MADRID VIA MADRID',
128    'MADRID VIA MADRID',
129    'MADRID VIA MADRID',
130    'MADRID VIA MADRID',
131    'MADRID VIA MADRID',
132    'MADRID VIA MADRID',
133    'MADRID VIA MADRID',
134    'MADRID VIA MADRID',
135    'MADRID VIA MADRID',
136    'MADRID VIA MADRID',
137    'MADRID VIA MADRID',
138    'MADRID VIA MADRID',
139    'MADRID VIA MADRID',
140    'MADRID VIA MADRID',
141    'MADRID VIA MADRID',
142    'MADRID VIA MADRID',
143    'MADRID VIA MADRID',
144    'MADRID VIA MADRID',
145    'MADRID VIA MADRID',
146    'MADRID VIA MADRID',
147    'MADRID VIA MADRID',
148    'MADRID VIA MADRID',
149    'MADRID VIA MADRID',
150    'MADRID VIA MADRID',
151    'MADRID VIA MADRID',
152    'MADRID VIA MADRID',
153    'MADRID VIA MADRID',
154    'MADRID VIA MADRID',
155    'MADRID VIA MADRID',
156    'MADRID VIA MADRID',
157    'MADRID VIA MADRID',
158    'MADRID VIA MADRID',
159    'MADRID VIA MADRID',
160    'MADRID VIA MADRID',
161    'MADRID VIA MADRID',
162    'MADRID VIA MADRID',
163    'MADRID VIA MADRID',
164    'MADRID VIA MADRID',
165    'MADRID VIA MADRID',
166    'MADRID VIA MADRID',
167    'MADRID VIA MADRID',
168    'MADRID VIA MADRID',
169    'MADRID VIA MADRID',
170    'MADRID VIA MADRID',
171    'MADRID VIA MADRID',
172    'MADRID VIA MADRID',
173    'MADRID VIA MADRID',
174    'MADRID VIA MADRID',
175    'MADRID VIA MADRID',
176    'MADRID VIA MADRID',
177    'MADRID VIA MADRID',
178    'MADRID VIA MADRID',
179    'MADRID VIA MADRID',
180    'MADRID VIA MADRID',
181    'MADRID VIA MADRID',
182    'MADRID VIA MADRID',
183    'MADRID VIA MADRID',
184    'MADRID VIA MADRID',
185    'MADRID VIA MADRID',
186    'MADRID VIA MADRID',
187    'MADRID VIA MADRID',
188    'MADRID VIA MADRID',
189    'MADRID VIA MADRID',
190    'MADRID VIA MADRID',
191    'MADRID VIA MADRID',
192    'MADRID VIA MADRID',
193    'MADRID VIA MADRID',
194    'MADRID VIA MADRID',
195    'MADRID VIA MADRID',
196    'MADRID VIA MADRID',
197    'MADRID VIA MADRID',
198    'MADRID VIA MADRID',
199    'MADRID VIA MADRID',
200    'MADRID VIA MADRID',
201    'MADRID VIA MADRID',
202    'MADRID VIA MADRID',
203    'MADRID VIA MADRID',
204    'MADRID VIA MADRID',
205    'MADRID VIA MADRID',
206    'MADRID VIA MADRID',
207    'MADRID VIA MADRID',
208    'MADRID VIA MADRID',
209    'MADRID VIA MADRID',
210    'MADRID VIA MADRID',
211    'MADRID VIA MADRID',
212    'MADRID VIA MADRID',
213    'MADRID VIA MADRID',
214    'MADRID VIA MADRID',
215    'MADRID VIA MADRID',
216    'MADRID VIA MADRID',
217    'MADRID VIA MADRID',
218    'MADRID VIA MADRID',
219    'MADRID VIA MADRID',
220    'MADRID VIA MADRID',
221    'MADRID VIA MADRID',
222    'MADRID VIA MADRID',
223    'MADRID VIA MADRID',
224    'MADRID VIA MADRID',
225    'MADRID VIA MADRID',
226    'MADRID VIA MADRID',
227    'MADRID VIA MADRID',
228    'MADRID VIA MADRID',
229    'MADRID VIA MADRID',
230    'MADRID VIA MADRID',
231    'MADRID VIA MADRID',
232    'MADRID VIA MADRID',
233    'MADRID VIA MADRID',
234    'MADRID VIA MADRID',
235    'MADRID VIA MADRID',
236    'MADRID VIA MADRID',
237    'MADRID VIA MADRID',
238    'MADRID VIA MADRID',
239    'MADRID VIA MADRID',
240    'MADRID VIA MADRID',
241    'MADRID VIA MADRID',
242    'MADRID VIA MADRID',
243    'MADRID VIA MADRID',
244    'MADRID VIA MADRID',
245    'MADRID VIA MADRID',
246    'MADRID VIA MADRID',
247    'MADRID VIA MADRID',
248    'MADRID VIA MADRID',
249    'MADRID VIA MADRID',
250    'MADRID VIA MADRID',
251    'MADRID VIA MADRID',
252    'MADRID VIA MADRID',
253    'MADRID VIA MADRID',
254    'MADRID VIA MADRID',
255    'MADRID VIA MADRID',
256    'MADRID VIA MADRID',
257    'MADRID VIA MADRID',
258    'MADRID VIA MADRID',
259    'MADRID VIA MADRID',
260    'MADRID VIA MADRID',
261    'MADRID VIA MADRID',
262    'MADRID VIA MADRID',
263    'MADRID VIA MADRID',
264    'MADRID VIA MADRID',
265    'MADRID VIA MADRID',
266    'MADRID VIA MADRID',
267    'MADRID VIA MADRID',
268    'MADRID VIA MADRID',
269    'MADRID VIA MADRID',
270    'MADRID VIA MADRID',
271    'MADRID VIA MADRID',
272    'MADRID VIA MADRID',
273    'MADRID VIA MADRID',
274    'MADRID VIA MADRID',
275    'MADRID VIA MADRID',
276    'MADRID VIA MADRID',
277    'MADRID VIA MADRID',
278    'MADRID VIA MADRID',
279    'MADRID VIA MADRID',
280    'MADRID VIA MADRID',
281    'MADRID VIA MADRID',
282    'MADRID VIA MADRID',
283    'MADRID VIA MADRID',
284    'MADRID VIA MADRID',
285    'MADRID VIA MADRID',
286    'MADRID VIA MADRID',
287    'MADRID VIA MADRID',
288    'MADRID VIA MADRID',
289    'MADRID VIA MADRID',
290    'MADRID VIA MADRID',
291    'MADRID VIA MADRID',
292    'MADRID VIA MADRID',
293    'MADRID VIA MADRID',
294    'MADRID VIA MADRID',
295    'MADRID VIA MADRID',
296    'MADRID VIA MADRID',
297    'MADRID VIA MADRID',
298    'MADRID VIA MADRID',
299    'MADRID VIA MADRID',
300    'MADRID VIA MADRID',
301    'MADRID VIA MADRID',
302    'MADRID VIA MADRID',
303    'MADRID VIA MADRID',
304    'MADRID VIA MADRID',
305    'MADRID VIA MADRID',
306    'MADRID VIA MADRID',
307    'MADRID VIA MADRID',
308    'MADRID VIA MADRID',
309    'MADRID VIA MADRID',
310    'MADRID VIA MADRID',
311    'MADRID VIA MADRID',
312    'MADRID VIA MADRID',
313    'MADRID VIA MADRID',
314    'MADRID VIA MADRID',
315    'MADRID VIA MADRID',
316    'MADRID VIA MADRID',
317    'MADRID VIA MADRID',
318    'MADRID VIA MADRID',
319    'MADRID VIA MADRID',
320    'MADRID VIA MADRID',
321    'MADRID VIA MADRID',
322    'MADRID VIA MADRID',
323    'MADRID VIA MADRID',
324    'MADRID VIA MADRID',
325    'MADRID
```

```

100         if (_isLoadingRoute)
101             Container(
102                 color: Colors.black26,
103                 child: const Center(
104                     ),
105             ),
106         ),
107     ),
108     if (!_isPassengerView)
109     BottomModal(
110         isExpanded: _isExpanded,
111         onExpandToggle: () {
112             setState(() => _isExpanded = !_isExpanded);
113         },
114         onVerticalDragEnd: _onVerticalDragEnd,
115         routeController: _routeController,
116         passengersController: _passengersController,
117         routeFocusNode: _routeFocusNode,
118         passengersFocusNode: _passengersFocusNode,
119         autoCompleteSuggestion: _autoCompleteSuggestion,
120         onCompleteText: _onCompleteText,
121         onSubmit: _onSubmit,
122     ),
123     if (_isPassengerView)
124     PassengerWidget(
125         maxPassengers: _maxPassengers,
126         onClose: () => setState(() => _isPassengerView = false),
127     ),
128     // Top-left logout button (replace existing waybeop)
129     LogoutButton(
130         onPressed: () {
131             Navigator.of(context).pushAndRemoveUntil(
132                 MaterialPageRoute(builder: (_) => const LoginPage()),
133                 (route) => false,
134             );
135         },
136     ),
137     // Recenter button positioned above the modal/passenger area at top-right
138     Builder(builder: (ctx) {
139         final screenHeight = MediaQuery.of(context).size.height;
140         // modal height when not in passenger view
141         final modalHeight = _isExpanded ? screenHeight * 0.38 : screenHeight * 0.09;
142         // when passenger view is active, approximate the passenger widget height
143         // PassengerWidget uses sidelize-76 and midsize-120; choose a conservative value
144         final passengerHeight = 120.0;
145
146         final double bottomOffset = !_isPassengerView
147             ? modalHeight + 12.0
148             : (MediaQuery.of(context).padding.bottom + 16.0) + passengerHeight + 12.0;
149
150         return RecenterButton(
151             bottom: bottomOffset,
152             right: 20,
153             onPressed: _recenterToluser,
154         );
155     },
156 ),
157 ),
158 ),
159 ),
160 ),
161 )
162 )

```

Commuter.dart

[illegible]

```

90 // Calculate distance of a polyline to kilometers
91 double _calculatePolyDistance(LatLngBounds points) {
92     if (points.length < 2) return 0.0;
93     double totalArea = 0.0;
94     for (int i = 0; i < points.length - 2; i++) {
95         totalArea += _calculateDistanceBetween(points[i], points[i + 2]);
96     }
97     return totalArea / 1000.0; // Convert to kilometers
98 }
99
100 // Extract route name from polyline ID and convert to display format
101 // Converts "hugs-alays" to "Hugs Alays" (camel case with spaces)
102 String _extractRouteName(String polylineId) {
103     // polylineId tagged to identify route name partitions (e.g., "jade-valley-0")
104     String parts = polylineId.split("polyId=")[1];
105
106     // Remove the last part if it's a number (the index)
107     if (parts.startsWith "0" || parts.startsWith "1") {
108         parts = parts.substring(0, parts.length() - 1);
109     }
110
111     // Split all remaining parts and capitalize each word
112     final sb = new StringBuilder();
113     final words = parts.split(" ");
114     final capitalizedWords = words.map((word) {
115         if (word.isEmpty()) return word;
116         return word[0].toUpperCase() + word.substring(1).toLowerCase();
117     }).toList();
118
119     return capitalizedWords.join(" ");
120 }
121
122 // Load a route by name using the conductor map service.
123 Future<Route> loadRoute(String name) async {
124     final layers = await ConductorRoadsHelper.getLayersByRouteName(name);
125     return () {
126         _polyline = layers.polyline.toLatLngList();
127         _polygon = layers.polygon.toLatLngList();
128         _markers = layers.markers.toList();
129     };
130 }
131
132 if (layers.polyline.isNotEmpty || _mapController != null) {
133     // Fit bounds to first polyline
134     final points = layers.polyline.first.points;
135     double minLat = points.first.latitude, minLat = points.first.latitude;
136     double minLng = points.first.longitude, minLng = points.first.longitude;
137     for (final p in points) {
138         minLat = p.latitude < minLat ? p.latitude : minLat;
139         minLat = p.latitude > minLat ? p.latitude : minLat;
140         minLng = p.longitude < minLng ? p.longitude : minLng;
141         minLng = p.longitude > minLng ? p.longitude : minLng;
142     }
143     final bounds = LatLngBounds(southwest: LatLng(minLat, minLng), northeast: LatLng(maxLat, maxLng));
144     _mapController?.animateCamera(CameraUpdate.newLatLngBounds(bounds, 40));
145 }
146
147 @override
148 Widget build(BuildContext context) {
149     return Scaffold(
150         body: Stack(
151             children: [
152                 MapWidget(
153                     mapController: _mapController,
154                     initialCameraPosition: initialPosition,
155                     onTap: () {
156                         // Handle map tap
157                     },
158                     onMapCreated: () {
159                         // Handle map creation
160                     },
161                     onMapDestroyed: () {
162                         // Handle map destruction
163                     },
164                     onMapLoaded: () {
165                         // Handle map loading
166                     },
167                     onMapReady: () {
168                         // Handle map ready
169                     },
170                     onMapUpdated: () {
171                         // Handle map update
172                     },
173                     onMapZoomed: () {
174                         // Handle map zoom
175                     },
176                     onMapZoomedOut: () {
177                         // Handle map zoom out
178                     },
179                     onMapZoomedIn: () {
180                         // Handle map zoom in
181                     },
182                     onMapZoomedReset: () {
183                         // Handle map zoom reset
184                     },
185                     onMapZoomedMax: () {
186                         // Handle map zoom max
187                     },
188                     onMapZoomedMin: () {
189                         // Handle map zoom min
190                     },
191                     onMapZoomedDefault: () {
192                         // Handle map zoom default
193                     },
194                     onMapZoomedCancel: () {
195                         // Handle map zoom cancel
196                     },
197                     onMapZoomedDismiss: () {
198                         // Handle map zoom dismiss
199                     },
200                     onMapZoomedError: () {
201                         // Handle map zoom error
202                     },
203                     onMapZoomedSuccess: () {
204                         // Handle map zoom success
205                     },
206                     onMapZoomedWarning: () {
207                         // Handle map zoom warning
208                     },
209                     onMapZoomedInfo: () {
210                         // Handle map zoom info
211                     },
212                     onMapZoomedDebug: () {
213                         // Handle map zoom debug
214                     },
215                     onMapZoomedLog: () {
216                         // Handle map zoom log
217                     },
218                     onMapZoomedPrint: () {
219                         // Handle map zoom print
220                     },
221                     onMapZoomedSave: () {
222                         // Handle map zoom save
223                     },
224                     onMapZoomedLoad: () {
225                         // Handle map zoom load
226                     },
227                     onMapZoomedClear: () {
228                         // Handle map zoom clear
229                     },
230                     onMapZoomedResetAll: () {
231                         // Handle map zoom reset all
232                     },
233                     onMapZoomedResetDefault: () {
234                         // Handle map zoom reset default
235                     },
236                     onMapZoomedResetMax: () {
237                         // Handle map zoom reset max
238                     },
239                     onMapZoomedResetMin: () {
240                         // Handle map zoom reset min
241                     },
242                     onMapZoomedResetDefault2: () {
243                         // Handle map zoom reset default 2
244                     },
245                     onMapZoomedResetMax2: () {
246                         // Handle map zoom reset max 2
247                     },
248                     onMapZoomedResetMin2: () {
249                         // Handle map zoom reset min 2
250                     },
251                     onMapZoomedResetDefault3: () {
252                         // Handle map zoom reset default 3
253                     },
254                     onMapZoomedResetMax3: () {
255                         // Handle map zoom reset max 3
256                     },
257                     onMapZoomedResetMin3: () {
258                         // Handle map zoom reset min 3
259                     },
260                     onMapZoomedResetDefault4: () {
261                         // Handle map zoom reset default 4
262                     },
263                     onMapZoomedResetMax4: () {
264                         // Handle map zoom reset max 4
265                     },
266                     onMapZoomedResetMin4: () {
267                         // Handle map zoom reset min 4
268                     },
269                     onMapZoomedResetDefault5: () {
270                         // Handle map zoom reset default 5
271                     },
272                     onMapZoomedResetMax5: () {
273                         // Handle map zoom reset max 5
274                     },
275                     onMapZoomedResetMin5: () {
276                         // Handle map zoom reset min 5
277                     },
278                     onMapZoomedResetDefault6: () {
279                         // Handle map zoom reset default 6
280                     },
281                     onMapZoomedResetMax6: () {
282                         // Handle map zoom reset max 6
283                     },
284                     onMapZoomedResetMin6: () {
285                         // Handle map zoom reset min 6
286                     },
287                     onMapZoomedResetDefault7: () {
288                         // Handle map zoom reset default 7
289                     },
290                     onMapZoomedResetMax7: () {
291                         // Handle map zoom reset max 7
292                     },
293                     onMapZoomedResetMin7: () {
294                         // Handle map zoom reset min 7
295                     },
296                     onMapZoomedResetDefault8: () {
297                         // Handle map zoom reset default 8
298                     },
299                     onMapZoomedResetMax8: () {
300                         // Handle map zoom reset max 8
301                     },
302                     onMapZoomedResetMin8: () {
303                         // Handle map zoom reset min 8
304                     },
305                     onMapZoomedResetDefault9: () {
306                         // Handle map zoom reset default 9
307                     },
308                     onMapZoomedResetMax9: () {
309                         // Handle map zoom reset max 9
310                     },
311                     onMapZoomedResetMin9: () {
312                         // Handle map zoom reset min 9
313                     },
314                     onMapZoomedResetDefault10: () {
315                         // Handle map zoom reset default 10
316                     },
317                     onMapZoomedResetMax10: () {
318                         // Handle map zoom reset max 10
319                     },
320                     onMapZoomedResetMin10: () {
321                         // Handle map zoom reset min 10
322                     },
323                     onMapZoomedResetDefault11: () {
324                         // Handle map zoom reset default 11
325                     },
326                     onMapZoomedResetMax11: () {
327                         // Handle map zoom reset max 11
328                     },
329                     onMapZoomedResetMin11: () {
330                         // Handle map zoom reset min 11
331                     },
332                     onMapZoomedResetDefault12: () {
333                         // Handle map zoom reset default 12
334                     },
335                     onMapZoomedResetMax12: () {
336                         // Handle map zoom reset max 12
337                     },
338                     onMapZoomedResetMin12: () {
339                         // Handle map zoom reset min 12
340                     },
341                     onMapZoomedResetDefault13: () {
342                         // Handle map zoom reset default 13
343                     },
344                     onMapZoomedResetMax13: () {
345                         // Handle map zoom reset max 13
346                     },
347                     onMapZoomedResetMin13: () {
348                         // Handle map zoom reset min 13
349                     },
350                     onMapZoomedResetDefault14: () {
351                         // Handle map zoom reset default 14
352                     },
353                     onMapZoomedResetMax14: () {
354                         // Handle map zoom reset max 14
355                     },
356                     onMapZoomedResetMin14: () {
357                         // Handle map zoom reset min 14
358                     },
359                     onMapZoomedResetDefault15: () {
360                         // Handle map zoom reset default 15
361                     },
362                     onMapZoomedResetMax15: () {
363                         // Handle map zoom reset max 15
364                     },
365                     onMapZoomedResetMin15: () {
366                         // Handle map zoom reset min 15
367                     },
368                     onMapZoomedResetDefault16: () {
369                         // Handle map zoom reset default 16
370                     },
371                     onMapZoomedResetMax16: () {
372                         // Handle map zoom reset max 16
373                     },
374                     onMapZoomedResetMin16: () {
375                         // Handle map zoom reset min 16
376                     },
377                     onMapZoomedResetDefault17: () {
378                         // Handle map zoom reset default 17
379                     },
380                     onMapZoomedResetMax17: () {
381                         // Handle map zoom reset max 17
382                     },
383                     onMapZoomedResetMin17: () {
384                         // Handle map zoom reset min 17
385                     },
386                     onMapZoomedResetDefault18: () {
387                         // Handle map zoom reset default 18
388                     },
389                     onMapZoomedResetMax18: () {
390                         // Handle map zoom reset max 18
391                     },
392                     onMapZoomedResetMin18: () {
393                         // Handle map zoom reset min 18
394                     },
395                     onMapZoomedResetDefault19: () {
396                         // Handle map zoom reset default 19
397                     },
398                     onMapZoomedResetMax19: () {
399                         // Handle map zoom reset max 19
400                     },
401                     onMapZoomedResetMin19: () {
402                         // Handle map zoom reset min 19
403                     },
404                     onMapZoomedResetDefault20: () {
405                         // Handle map zoom reset default 20
406                     },
407                     onMapZoomedResetMax20: () {
408                         // Handle map zoom reset max 20
409                     },
410                     onMapZoomedResetMin20: () {
411                         // Handle map zoom reset min 20
412                     },
413                     onMapZoomedResetDefault21: () {
414                         // Handle map zoom reset default 21
415                     },
416                     onMapZoomedResetMax21: () {
417                         // Handle map zoom reset max 21
418                     },
419                     onMapZoomedResetMin21: () {
420                         // Handle map zoom reset min 21
421                     },
422                     onMapZoomedResetDefault22: () {
423                         // Handle map zoom reset default 22
424                     },
425                     onMapZoomedResetMax22: () {
426                         // Handle map zoom reset max 22
427                     },
428                     onMapZoomedResetMin22: () {
429                         // Handle map zoom reset min 22
430                     },
431                     onMapZoomedResetDefault23: () {
432                         // Handle map zoom reset default 23
433                     },
434                     onMapZoomedResetMax23: () {
435                         // Handle map zoom reset max 23
436                     },
437                     onMapZoomedResetMin23: () {
438                         // Handle map zoom reset min 23
439                     },
440                     onMapZoomedResetDefault24: () {
441                         // Handle map zoom reset default 24
442                     },
443                     onMapZoomedResetMax24: () {
444                         // Handle map zoom reset max 24
445                     },

```


Curriculum Vitae



Kieffer Lance A. Devera

Deca Home Esperanza, Blk 46 lot 28 St. Fluorite,
Tigatto, Davao City, 8000
(+63)9777000318
klDevera@mcm.edu.ph

Personal Statement

I am a recent graduate of Mapúa Malayan Colleges Mindanao with a Bachelor of Science in Computer Science, I am excited to apply my knowledge, creativity, and commitment to a workplace. My academic path has provided me with strong technical skills in programming, research, and software design, a profound understanding of the importance of teamwork, and constant improvement. I am now looking for a place where I can use what I've learned, develop as a young professional and make a meaningful contribution to a progressive enterprise. Motivated, flexible, and technology enthusiast, I am willing to contribute from day one.

Skills

- Programming Languages
Skilled in Python, Java, and C++, having worked on building academic projects and solving real-world problems using code. Familiar with procedural and object-oriented programming styles.
- Software Tools
Proficient with Visual Studio Code for coding and debugging, and Microsoft Office applications for developing professional reports, presentations, and documentation.
- Soft Skills
Team oriented and communicative, effective in group projects. Organized and able to manage time to meet deadlines and handle several tasks at a time.

Education Attainment

2024 – 2028	Mapúa Malayan Colleges Mindanao Bachelor of Science in Computer Science Davao City, Philippines
2022 – 2024	Colegio de San Ignacio Senior High School (STEM Strand) Davao City, Philippines

2018 – 2022	Colegio de San Ignacio Junior High School Davao City, Philippines
-------------	--

Work Experience

2022-2022	Activations Advertising Inc. Davao City Freelance Encoder
2024-2025	Coding Organization for Digital Excellence (C.O.D.E.) Mapúa Malayan Colleges Mindanao Head of Operations Officer
2024-2025	Computing Students Society (CSS) Mapúa Malayan Colleges Mindanao Project Development Committee Member

Certifications

2024	C++ Lab: Crafting Code in the Digital Age Certificate of Participation
2025	Visual Studio Code Workshop Certificate of Participation



Josef Dave R. Llorente

Lot3A, Blk3, Phase 3, El Rio, Ilang-ilang St., Davao City, 8000

(+63) 9923665633

josefdave.llorente@gmail.com

Personal Statement

I am a recent Computer Science graduate from Mapua Malayan Colleges Mindanao with knowledge and experience in data analysis, statistics, and programming. I am eager to join All-Around Business, Inc. as a Data Analyst to help turn data into clear and useful insights. I am dedicated, detail-oriented, and always willing to learn and improve.

Skills

Problem Analysis

Able to break down complex issues into smaller, manageable parts to identify effective and logical solutions. Skilled in recognizing patterns, root causes, and dependencies to guide problem-solving with clarity and purpose.

Simplifying Complexity

Capable of understanding intricate systems or data and presenting them in a clear, accessible way. Adept at translating complex concepts for better communication, collaboration, and decision-making across teams.

Team Collaboration

Works well in group settings by contributing ideas, sharing responsibilities, and supporting team goals.

Educational Attainment

2024 - 2027	Bachelor of science in Computer Science Mapua Malayan Colleges Mindanao McArthur Highway, Matina, Davao City
2022 - 2024	STEM Strand Graduate University of Immaculate Conception Bajada, Davao City
2018 - 2022	STEM Strand Graduate Digos City National High School Makar Road, Digos City

Work Experience

2024 - Present	Newwave Davao City Social Media Manager
	Newwave Davao City App Developer/Technical Assistant
2022 - 2023	Office of Dr. Jean Diar R. Llorente Davao City Secretary

Certifications

2025	CODECHUM: C++ Programming 1
2024	C++ Lab: Crafting Code in the Digital Age Blockchain Campus Conference
	First Line of Defense: AWS Account Security 101 with IAM
	Champion, Research Poster Defense