University of 20 août 1955 Skikda
Faculty of science
Department of Computer Science

A project to obtain a master's degree + a certificate for a startup project within the framework of the ministerial decision 1275

Smart Mobility Application: Improving public transportation operations using artificial intelligence.

Authors

Mohamed Nabil BOURBIA
Sabri BOUFARES
Said Fadhel Hasen NADJI

Jury

President: /

Supervisor: Dr. Samira HAZMOUNE

Co-supervisor: Dr. Fateh BOUGAMOUZA

Examiner: Dr. Hanene BOURRICHE

BMC Expert: /

Commercial Partner: /

Submission date: July 2024

Dedication

We dedicate the humble fruit of our efforts to those who were the cause of our success.

To our parents, may Allah extend their lives.

To our brothers and sisters, each by name.

To all our family and relatives.

To all those who taught us a letter throughout our educational journey from elementary to university, our esteemed teachers.

To our colleagues who shared our studies with us.

To all who helped us by word and deed and were a support to us in completing this work.

To all who strive and struggle for the sake of education.

To all who have love, respect, and appreciation for us.

Acknowledgment

Our deepest gratitude and praise be to Allah Almighty, who has granted us the completion of this humble work. All praise be to Him as befits the majesty of His Face and the greatness of His Power.

We extend our deepest gratitude to our supervisor, Dr. Samira BOUGAMOUZA, and Dr. Fateh BOUGAMOUZA, who accompanied us throughout this project and enriched us with valuable information and advice. We pray to Allah Almighty to guide their steps and fulfill their wishes. May Allah reward them with all the best on our behalf.

We also thank our deepest gratitude to all the officials and workers of the incubator and the entrepreneurial center of the University of 20 August 1955 Skikda for the assistance and facilitation provided. May Allah reward them with all the best on our behalf.

To all the professors of the Department of Computer Science, each in their own position, and to all who helped us from near or far in completing this humble work.

Last but not least, we extend our thanks to the esteemed members of the examination committee for accepting to judge this thesis.

Abstract

The spread of urbanization and the increase in population density in urban areas have highlighted the importance of public transportation systems. With the growing population in cities, the demand for public transportation options that are easy and efficient has also increased. However, traditional public transportation services often fail to meet the dynamic needs of contemporary passengers, who require accurate and timely information for their daily commutes to avoid unnecessary time and effort.

Therefore, the development of an advanced application that enables users to access public transportation services easily and seamlessly without the need for inquiries or manual assistance is crucial.

This master's thesis focuses on creating an application that facilitates urban transportation and mobility without the need for inquiries or assistance from others, utilizing artificial intelligence and reinforcement learning techniques.

Keywords: Public transportation, Artificial intelligence, Mobile applications, Graph Neural Networks, Deep Reinforcement Learning, Graph theory, Optimisation, Smart mobility.

ملخص

أدى انتشار التحضر وزيادة الكثافة السكانية في المناطق الحضرية إلى تسليط الضوء على أهمية نظم النقل العام. ومع تزايد عدد سكان المدن، ازداد الطلب على خيارات النقل العام التي تتميز بالسهولة والفعالية. ومع ذلك، فإن خدمات النقل العام التقليدية غالبا ما تفشل في تلبية الإحتياجات الديناميكية للركاب المعاصرين، الذين يحتاجون إلى معلومات دقيقة وفي الوقت المناسب للتنقل عبر مساراتهم اليومية، مما يؤدي إلى تكبيدهم الوقت والجهد.

وبناءً على ذلك، يعتبر تطوير تطبيق متقدم يمكِّن المستخدمين من الوصول إلى خدمات النقل العام بسهولة وسلاسة دون الحاجة إلى استفسارات أو مساعدة يدوية أمراً بالغ الأهمية.

في هذه الدراسة لرسالة الماستر، نركز على إنشاء تطبيق يسهم في تسهيل عملية النقل والتنقل في المدن دون الحاجة إلى الاستفسارات أو مساعدة الآخرين، وذلك باستخدام تقنيات الذكاء الاصطناعي والتعلم المعزز.

الكلمات المفتاحية: النقل العام، الذكاء الاصطناعي، تطبيق الهاتف، الشبكات العصبية البيانية، التعلم العميق المعزز، التحسين، التنقل الذكي.

Table of Contents

List of Figures

General introduction

The proliferation of urbanization and the increasing population density in metropolitan areas have accentuated the importance of efficient public transportation systems As cities grow the demand for accessible reliable and user-friendly transportation options has surged Traditional public transport services through extensive often fall short in addressing the dynamic needs of modern commuters who seek timely and precise information to navigate through their daily routes The development of an advanced system that enables users to access and utilize public transportation services seamlessly without the need for manual inquiries or assistance has become paramount.

This project aims to develop such an advanced system leveraging cutting-edge technologies like real-time data analytics and machine learning algorithms to provide users with accurate and up-to-date information on public transportation options including routes stops and schedules by harnessing these technologies the proposed system will transform the public transportation experience making it more intuitive and efficient for users.

Public transportation plays a critical role in urban mobility offering a sustainable and efficient alternative to private vehicle use It helps reduce traffic congestion lowers carbon emissions and promotes equitable access to various city amenities However the effectiveness of public transportation systems is often hampered by several challenges including inconsistent schedules lack of real-time information and the complexity of navigating multiple transit routes.

The traditional methods of obtaining information about public transportation such as static schedules posted at stops or manual inquiries through customer service are inadequate in meeting the expectations of today's tech-savvy commuters These methods are not only time-consuming but also fail to provide the dynamic and immediate updates needed to cope with disruptions or changes in transit services.

The advent of real-time data analytics and machine learning has opened new avenues for enhancing public transportation systems Real-time data analytics involves processing and analyzing data as it is generated allowing for instantaneous insights and decision-making In the context of public transportation real-time data can include information on vehicle locations traffic conditions and passenger loads among other parameters.

Machine learning a subset of artificial intelligence involves training algorithms to recognize patterns and make predictions based on historical and real-time data These technologies can be employed to forecast transit schedules optimize routes and provide personalized recommenda-

tions to users based on their travel patterns The proposed system will integrate real-time data analytics and machine learning to deliver a comprehensive and user-centric public transportation solution The system will consist of several key components. The system will gather data from various sources including GPS devices on public transport vehicles traffic management systems and user inputs This data will be integrated into a centralized database for processing. Using advanced data analytics techniques, the system will process the collected data to provide real-time updates on transit routes vehicle locations and schedule changes This information will be continuously updated and made accessible to users through a user-friendly interface.

The system will employ machine learning algorithms to analyze historical and real-time data enabling it to predict transit schedules and optimize routes These algorithms will learn from patterns in the data improving their accuracy over time. A crucial component of the system is its user interface designed to be intuitive and accessible Users will be able to input their desired destinations and receive detailed information on the best routes estimated arrival times and any potential delays or disruptions The interface will also support notifications and alerts to keep users informed of any significant changes.

The implementation of this advanced public transportation system will offer several significant benefits. By providing real-time information and personalized recommendations, the system will make navigating public transportation more straightforward and efficient for users. This will reduce the time spent waiting for buses or trains and minimize the uncertainty associated with transit schedules. Additionally, transit authorities will be able to use the insights generated by the system to optimize their operations. This includes adjusting schedules based on demand, rerouting vehicles to avoid congestion, and improving overall service reliability. By making public transportation more attractive and convenient, the system will encourage more people to use it, thereby reducing the number of private vehicles on the road. This will contribute to lower traffic congestion and decreased environmental impact. Furthermore, the system will be designed to cater to all users, including those with disabilities. Features such as audio alerts and easy-to-navigate interfaces will ensure that everyone can benefit from the improved public transportation services.

This manuscript is divided into four chapters. Chapter One, titled ”AI Techniques in Optimization,” discusses graph theory, metaheuristics, and machine learning, specifically reinforcement learning and graph neural networks. Chapter Two, titled ”Application of AI in Public Transportation,” covers topics such as public transportation, graph neural networks, and deep rein-

forcement learning. Chapter Three delves into the conception, presenting various diagrams. Finally, Chapter Four focuses on the working environment and tools, detailing the tools used in development and providing screenshots of the developed application.

Chapter 1

Artificial intelligence techniques for optimization

1.1 Introduction

Chapter 1 serves as a foundational exploration into the dynamic intersection of artificial intelligence (AI) and optimization techniques. In today's rapidly evolving technological landscape, the demand for efficient problem-solving methodologies spans across diverse fields ranging from logistics and finance to engineering and beyond. At the heart of this demand lies the quest for optimal solutions that not only streamline operations but also maximize outcomes in terms of cost-effectiveness, resource utilization, and performance enhancement.

This chapter embarks on a comprehensive journey through the theoretical frameworks and practical applications of AI-driven optimization strategies. It begins by elucidating the fundamental concepts underpinning optimization, emphasizing its pivotal role in addressing complex decision-making scenarios where multiple variables and constraints interplay. By harnessing the power of AI, these traditional optimization challenges are approached with innovative methodologies that integrate advanced algorithms capable of learning, adapting, and evolving in response to dynamic environments.

The exploration then pivots towards a detailed examination of prominent AI techniques employed for optimization tasks. Genetic algorithms, inspired by natural selection processes, offer a heuristic approach to finding near-optimal solutions through iterative improvement and selection mechanisms. Neural networks, characterized by their ability to model complex nonlinear relationships, contribute significantly to pattern recognition and optimization tasks by

leveraging deep learning architectures. Concurrently, swarm intelligence algorithms, inspired by collective behavior observed in nature, harness decentralized decision-making to optimize solutions based on collaboration and information sharing among agents.

Through a synthesis of theoretical insights and practical examples, this chapter underscores the transformative impact of AI techniques on optimization across various domains. It illustrates how these methodologies empower decision-makers to navigate uncertainty and complexity with agility, precision, and scalability. Moreover, by examining case studies and real-world applications, the chapter illuminates the efficacy of AI-driven optimization in enhancing operational efficiency, driving innovation, and fostering competitive advantage in contemporary industries.

Looking forward, the chapter concludes with a forward-looking perspective on emerging trends and future directions in AI-driven optimization. It anticipates ongoing advancements in computational capabilities, algorithmic sophistication, and interdisciplinary collaboration, paving the way for novel applications and transformative outcomes in fields yet to be fully explored. As such, this introductory chapter sets the stage for a deeper exploration into the synergistic potential of AI and optimization, highlighting its role as a catalyst for innovation and progress in the digital era.

1.2 Graph theory

1.2.1 Definition

Graph theory in optimization is the application of graph theoretic concepts and tools to solve optimization problems. It utilizes graphs, which are mathematical structures consisting of vertices (nodes) connected by edges (links), to model real-world scenarios with inherent relationships. By expressing the problem as a graph, optimization techniques from graph theory can be employed to find the ”best” solution based on a defined objective function[1].

This ”best” solution can refer to various goals depending on the problem. Some common optimization objectives in graph theory include:

Minimizing distance (e.g., finding the shortest path between two points in a network) Maximizing flow (e.g., allocating resources efficiently across a network) Finding optimal matchings (e.g., pairing elements for maximum compatibility) The power of graph theory in optimization lies in its ability to represent complex relationships concisely. By leveraging properties

of graphs and efficient algorithms developed within graph theory, researchers and practitioners can tackle various optimization problems across diverse fields like:

- **Knowledge Representation**: Graphs can represent knowledge by mapping nodes to entities or concepts and edges to relationships between them. For example, in a medical knowledge graph, nodes could represent diseases, symptoms, and treatments, while edges denote associations like ”causes,” ”treats,” or ”is-a”.

- **Network Analysis**: Social networks are often modeled using graphs, where nodes represent individuals or entities, and edges signify relationships (friendships, collaborations). Algorithms can analyze these graphs to identify influential nodes, community structures, or information flow patterns.

- **Pathfinding and Optimization**: Graph algorithms, such as Dijkstra’s algorithm, are crucial for finding optimal paths in transportation networks (e.g., GPS navigation systems) or shortest paths in communication networks (e.g., internet routing).

- **Recommendation Systems**: Graph-based algorithms like collaborative filtering use user-item interaction data stored as a bipartite graph. Nodes represent users and items, and edges denote interactions (ratings, purchases). Recommendations are derived by exploring the graph structure to find connections between users with similar preferences.

- **Natural Language Processing**: Semantic networks use graphs to represent relationships between words or entities based on semantic similarity or co-occurrence in text. Nodes represent words or concepts, and edges signify semantic relationships (synonyms, hypernyms).

- **Cybersecurity**: Graph theory aids in detecting anomalies in network traffic by modeling connections between devices and identifying unusual patterns that may indicate security threats[2].

Graph theory’s versatility in AI lies in its ability to represent data and relationships in a way that facilitates efficient computation and reveals intricate patterns that might be obscured in other data structures. These techniques play a crucial role across diverse AI domains, from knowledge representation to network analysis and beyond.

1.2.2 Breadth-First Search (BFS)

1.2.2.1 Principle

Breadth-first search (BFS) is a fundamental algorithm for exploring tree or graph data structures. Unlike a spelunker venturing haphazardly into a cave, BFS follows a systematic approach. It meticulously visits all nodes at a specific level, ensuring no neighboring passages are missed, before venturing deeper into the structure. This level-by-level exploration prioritizes a thorough examination of the ”outer” regions before delving inward[3].

1.2.2.2 Key Characteristics

- **Exploration Strategy**: BFS explores nodes level by level, ensuring that it exhaustively searches all nodes at a given depth level before moving deeper.

- **Queue Mechanism**: It uses a first-in, first-out (FIFO) queue to manage the order of node exploration, ensuring nodes are processed in the order they are discovered.

1.2.2.3 Applications in AI and Graph Theory

- **Shortest Path Finding**:
  BFS is optimal for finding the shortest path in an unweighted graph because it explores nodes layer by layer. The first time a node is reached via BFS, it is reached using the shortest path possible.
  Example: In network routing, BFS can determine the shortest path between two nodes in a communication network where all connections have the same weight (e.g., hops or links).

- **Reachability and Connectivity Analysis**:
  BFS can determine which nodes are reachable from a given node and assess the connectivity of the graph. This is essential for understanding the structure and robustness of networks.
  Example: In social network analysis, BFS can identify clusters or communities by exploring nodes within a certain distance from a central node, helping to understand social connectivity patterns[4].

- **Optimizing Search Algorithms**:
  BFS serves as a foundational component in more complex algorithms, such as heuristic search methods like A* (A-star). A* combines BFS with heuristics to efficiently search for paths that are likely to lead to the optimal solution, especially in pathfinding problems with varying costs.
  Example: In robotics, BFS can be used to explore adjacent areas for path planning, ensuring that all possible routes are considered before making decisions.

1.2.2.4 Academic Perspective

In academia, BFS is studied extensively due to its simplicity, clarity, and foundational role in algorithmic design and analysis. Researchers focus on:

- **Algorithmic Complexity**: Analyzing BFS in terms of time complexity (O(V + E)) and space complexity (O(V)) where V is the number of vertices and E is the number of edges.

- **Applications in Various Fields**: Studying BFS applications across domains like network analysis, social sciences, biology (gene networks), and information retrieval.

- **Graph Theory Contributions**: BFS contributes to fundamental graph theory concepts such as connected components, bipartiteness testing, and cycle detection.

By mastering BFS and its applications, researchers and practitioners in AI can develop efficient solutions for diverse problems, leveraging its ability to systematically explore and analyze graph structures in various domains. This understanding fosters advancements in AI techniques ranging from network analysis to intelligent decision-making systems[5].

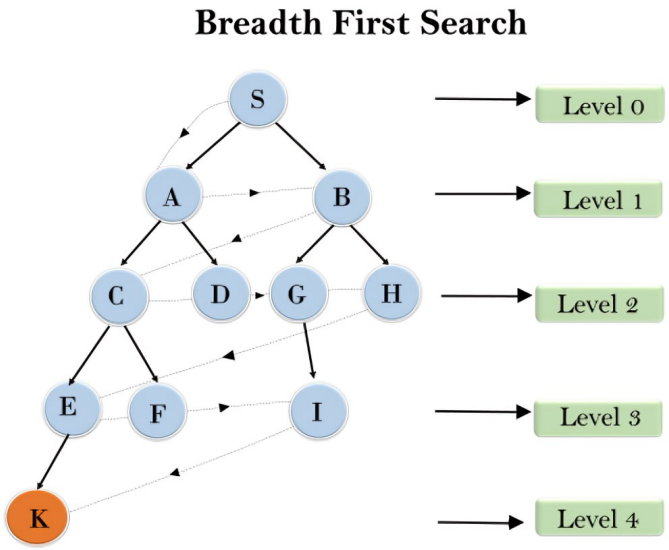


Figure 1.1: Breadth-first search (BFS) algorithm applied to a tree structure. [6]

1.2.3 Dijkstra's Algorithm

1.2.3.1 Principle

Dijkstra's Algorithm tackles the single-source shortest path problem in a weighted graph. Imagine a map (the graph) with cities (vertices) connected by roads (edges) with distances marked (weights). The algorithm efficiently finds the shortest route from a starting city (source vertex) to all other reachable cities. It works in stages, keeping track of "settled" cities where the shortest path is confirmed and "unsettled" cities where exploration continues. At each step, it picks the unsettled city closest (based on tentative distances) to the starting point, marks it as settled, and then examines all outgoing roads from that city. For each neighboring city, it compares the current shortest path estimate with the distance through the newly explored city. If the new path is shorter, the estimate is updated. This "relaxation" process ensures the algorithm progressively discovers and refines the shortest paths until all reachable cities are settled[7].

1.2.3.2 Key Characteristics

- **Priority Queue**: Dijkstra's Algorithm uses a priority queue to efficiently retrieve the node with the smallest tentative distance, making it suitable for large graphs with many nodes and edges.
- **Greedy Approach**: It employs a greedy strategy by always expanding the node with

the smallest tentative distance, ensuring that the shortest path to each node is found in a systematic manner[8]

1.2.3.3 Applications in AI and Graph Theory

- **Optimal Route Planning**: Transportation Networks: Dijkstra's Algorithm is pivotal in GPS navigation systems and transportation planning applications. It calculates the shortest route between two locations based on road distances and real-time traffic conditions.
  Example: In urban planning, Dijkstra's Algorithm can determine the quickest route for emergency services to reach a destination, taking into account traffic congestion and distance.

- **Resource Allocation**: Dijkstra's Algorithm can be adapted to allocate resources optimally in scenarios such as telecommunications networks or supply chain management, where minimizing costs or maximizing efficiency is crucial.
  Example: In telecommunications, it can determine the most efficient path for data packets to travel through a network based on current traffic and connection quality metrics.

- **Pathfinding in Robotics**: It plays a vital role in robotic path planning, where robots need to navigate through environments with obstacles while minimizing travel time or energy consumption.
  Example: Autonomous vehicles use variants of Dijkstra's Algorithm to plan safe and efficient routes based on real-time sensor data and environmental conditions[9]

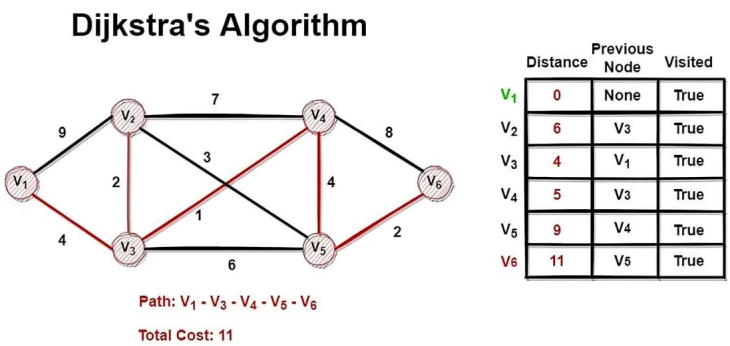


Figure 1.2: Dijkstra's Algorithm applied to a network of six nodes [10]

1.2.3.4 Academic Perspective

Dijkstra's Algorithm holds significant academic value due to its efficiency and versatility. Researchers delve into its time and space complexity (O((V + E) log V) and O(V) respectively, where V represents vertices and E represents edges). Furthermore, it contributes to core graph theory concepts like shortest path problems, single-source solutions, and graph traversal techniques. Academic exploration often focuses on variants and optimizations of the algorithm to tackle specific limitations or enhance performance in dynamic environments. By understanding and applying Dijkstra's Algorithm, researchers and AI practitioners can build advanced solutions for pathfinding, resource optimization, and decision-making in complex systems. Its role as a foundation in graph theory and AI ensures its continued relevance and fuels innovation across diverse applications[11]

1.2.4 A* Search Algorithm

1.2.4.1 Principle

A* Search Algorithm is a balance between efficiency and Accuracy. The A* (A-star) Search Algorithm stands out as a prominent technique for pathfinding and graph traversal. It leverages a combination of Dijkstra's algorithm and heuristic search to identify the most cost-effective path from a starting node to a designated goal node within the graph. This effectiveness hinges on the algorithm's ability to evaluate nodes based on two crucial cost components: g(n): This function represents the verified cost incurred from the starting node to the current node (n) under consideration. In essence, it captures the actual path traveled thus far.

H(n): This function embodies a heuristic estimate – an informed guess – regarding the cost remaining to reach the goal node from the current node (n). The accuracy of this heuristic significantly impacts the efficiency of A*. An ideal heuristic provides a close approximation to the true cost, guiding the search towards the goal and minimizing unnecessary exploration of irrelevant paths.

The core operation of A* involves a priority queue. This data structure prioritizes the exploration of nodes based on a combined cost function, denoted as f(n). This function is calculated as the sum of the two aforementioned cost components: f(n) = g(n) + h(n).

The effectiveness of A* hinges on the quality of the chosen heuristic function (h(n)). A precise heuristic that closely reflects the true cost to the goal fosters a more efficient search process by

directing the algorithm towards promising paths and deterring exploration of less likely routes.

1.2.4.2 Key Characteristics

- **Heuristic Function**: The choice of the heuristic function h(n).
  H(n) significantly impacts the performance of A*. A common heuristic is the Euclidean distance or Manhattan distance in grid-based maps.

- **Optimality**: A* is guaranteed to find the optimal path if the heuristic function h(n) is admissible (never overestimates the true cost) and consistent (satisfies the triangle inequality)[12].

1.2.4.3 Applications in AI and Graph Theory

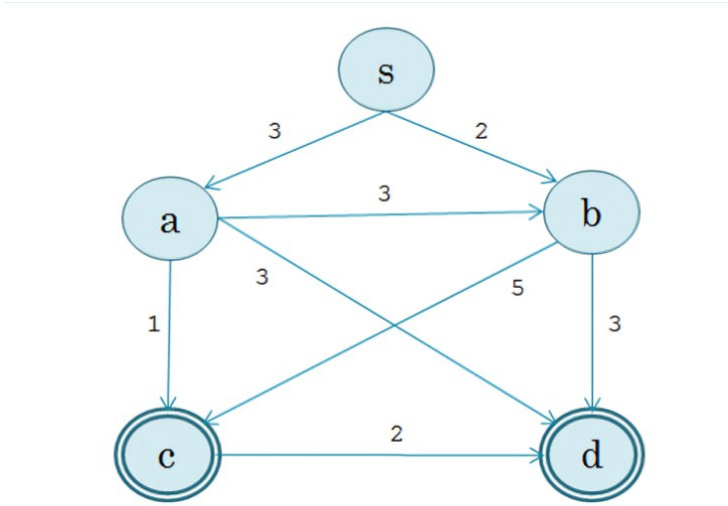
- **Pathfinding in Robotics**: Navigation: A* is extensively used in robotics for navigating environments with varying movement costs and obstacles. It ensures that robots find efficient paths while avoiding obstacles and adhering to movement constraints.
  Example: In autonomous mobile robots (AMRs) used in warehouses, A* helps determine optimal routes for picking and placing items, considering dynamic obstacles and varying floor conditions.

- **Game AI**: Real-time Strategy Games: A* is used for pathfinding in video games, allowing characters or units to navigate complex terrains efficiently.
  Example: In a real-time strategy game, A* enables units to find paths through battlefields, avoiding enemy positions and navigating around obstacles to reach strategic points.

- **Logistics and Transportation**: Route Optimization: A* is used for optimizing delivery routes in logistics, taking into account real-time traffic data, delivery priorities, and road conditions.
  Example: Delivery drones use A* to plan flight paths that minimize energy consumption while avoiding no-fly zones and other hazards.

- **Geographical Information Systems (GIS)**: Terrain Navigation: A* assists in finding paths across complex terrains in GIS applications, considering elevation changes, terrain types, and other geographical features.

Example: In hiking and outdoor navigation apps, A* helps users plan routes that avoid steep climbs and find the most scenic paths.



h(s)	h(a)	h(b)	h(c)	h(d)
1	3	3	0	0

Figure 1.3: The A* search algorithm applied to a graph [13]

1.2.4.4 Academic Perspective

Within the realm of academia, A* Search garners significant attention due to its captivating interplay between theoretical properties and practical applications. Researchers delve into its algorithmic complexity, understanding that this complexity hinges on the quality of the chosen heuristic function. In the worst-case scenario, A* exhibits a time complexity of $O(b^d)$, where ”b” represents the branching factor of the graph and ”d” signifies the depth of the solution path. This analysis provides valuable insights into the algorithm’s efficiency under various conditions.

A significant area of academic exploration focuses on crafting effective heuristic functions. These functions strive for a delicate balance between accuracy in estimating the cost to the goal and maintaining computational efficiency during the search process. Heuristics that are both admissible (guaranteed not to overestimate the cost) and consistent (always return the same estimate for a given node) are crucial for ensuring A*’s optimality, guaranteeing it finds the most

cost-effective path.

Academic inquiry extends beyond the core A* algorithm, venturing into the exploration of extensions and variants. These advancements encompass algorithms like weighted A* (WA*), iterative deepening A* (IDA*), and A* with dynamic replanning (D*). Each variant addresses specific limitations or enhances performance in dynamic environments or large-scale graphs, expanding A*'s reach and effectiveness.

By mastering A*, researchers and practitioners in AI gain the power to tackle intricate pathfinding and optimization problems. This proficiency elevates A* to the status of a cornerstone algorithm, finding applications in diverse fields like robotics, logistics, and many more. Its ability to efficiently navigate diverse environments and identify optimal paths underscores its significance in both theoretical computer science and real-world applications[14].

1.3 Metaheuristic techniques

1.3.1 Definition

Metaheuristic optimization is a subfield of optimization that utilizes metaheuristic algorithms to tackle complex optimization problems. Optimization problems are ubiquitous, encompassing diverse domains such as engineering design, economics, and even logistical tasks like holiday planning and internet routing. A core principle in optimization is achieving the optimal utility of available resources, considering constraints like time, money, and materials.

Finding the optimal solution, or even a very good sub-optimal solution (a solution close to the best but not necessarily perfect), can be a significant challenge in such scenarios. This is where metaheuristic optimization comes into play.

Metaheuristic algorithms provide a general framework for approaching these complex optimization problems. They don't guarantee finding the absolute optimum, but they aim to efficiently identify high-quality solutions within a reasonable timeframe[15].

1.3.2 Simulated Annealing (SA)

1.3.2.1 Principle

Simulated annealing draws inspiration from the physical process of annealing metals to achieve a low-energy, optimally ordered state. In the realm of optimization problems, it tackles com-

plex landscapes riddled with multiple local optima (seemingly best solutions in their immediate vicinity). Here's how it works: The algorithm iteratively explores the solution space, transitioning from one state (candidate solution) to another. Each state is assigned an energy value, which translates to "goodness" within the context of the optimization problem. A high temperature allows the algorithm to move between states with both lower (better) and higher (worse) energy. As the process progresses, the temperature is gradually reduced (cooling). This mechanism favors transitions towards states with lower energy, ultimately guiding the algorithm towards a good solution, potentially even the global optimum[16].
This controlled acceptance of worse solutions early on, akin to allowing uphill climbs during metal annealing, plays a crucial role. It helps the algorithm escape from local optima and explore the landscape more broadly, leading to the potential discovery of significantly better final solutions[17].

1.3.2.2 Key Characteristics

- **Exploration vs. Exploitation**: High initial temperature allows exploration; gradual cooling leads to exploitation.
- **Probability-Based Acceptance**: Even suboptimal solutions can be accepted with a certain probability, which decreases over time.

1.3.2.3 Applications

- **Pathfinding**: Optimizing routes in transportation networks.
  Example: In transportation networks, Simulated Annealing can be used to find optimal routes by initially exploring various paths and gradually focusing on the best-performing ones as the algorithm "cools down"[18].
- **Scheduling**: Allocating resources or tasks over time in manufacturing or computing.

1.3.3 Genetic Algorithms (GA)

1.3.3.1 Principle

Inspired by the process of natural selection, Genetic Algorithms (GAs) tackle optimization problems by mimicking biological evolution, excelling at finding optimal solutions in complex land-

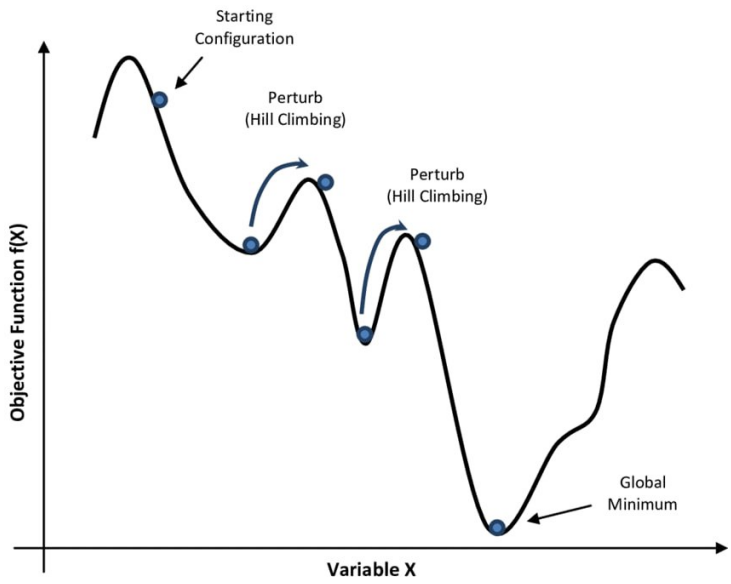


Figure 1.4: Simulated annealing objective function
[19]

scapes, especially when the relationships between variables and the desired outcome are unclear. The process begins with a population of candidate solutions (individuals), each encoded as a chromosome. These individuals are evaluated using a fitness function that measures how well they solve the optimization problem, with higher fitness scores indicating better solutions. Individuals are then selected for reproduction based on their fitness, with higher fitness increasing the chances of being chosen, following the principle of "survival of the fittest." Selected individuals undergo crossover, swapping portions of their chromosomes to create new offspring, fostering exploration of the search space and potential discovery of improved solutions. Mutations are introduced at a low probability to offspring's chromosomes to maintain genetic diversity and prevent the population from getting stuck in local optima. New offspring replace a portion of the less fit older population, and this cycle of evaluation, selection, crossover, mutation, and replacement iterates over generations. Over time, the population evolves towards better solutions, with the fittest individuals in the final generation representing high-quality solutions to the optimization problem[20].

1.3.3.2 Key Characteristics

- **Population-Based Search**: Maintains a diverse set of solutions.
- **Crossover and Mutation**: Combines and alters solutions to explore new areas of the search space.

1.3.3.3 Applications

- **Pathfinding**: Finding efficient routes in dynamic environments.
  Example: In robotic path planning, Genetic Algorithms can evolve a population of possible paths, combining the best ones and introducing mutations to avoid obstacles and find the shortest route[21].

- **Optimization**: Solving complex combinatorial problems like the Traveling Salesman Problem (TSP).

1.3.4 Particle Swarm Optimization (PSO)

1.3.4.1 Principle

Inspired by the collective behavior of swarming creatures like birds or fish, Particle Swarm Optimization (PSO) is an algorithm designed to solve optimization problems. It is particularly effective in navigating complex landscapes where traditional methods may struggle.
In PSO, a population of particles represents candidate solutions in the search space. Each particle has a position, indicating its current solution, and a velocity, indicating its movement direction. The fitness of each particle is assessed using a fitness function, which measures how well it solves the optimization problem.
Each particle records its best position (personal best) and the best position found by any particle in its neighborhood (social best). The particle's velocity is updated based on its current velocity, personal best, and social best. This update includes inertia (maintaining movement), attraction to its own best experience (exploitation), and attraction to the best solution found by its neighbors (exploration). The particle's position is then updated using the new velocity, moving it towards potentially better regions of the search space.

This process of evaluating fitness, updating velocity, and updating position repeats over many cycles. Throughout these iterations, the swarm of particles dynamically adjusts its search based on individual and collective experiences, progressively converging towards promising areas of the search space. This collaborative approach helps the swarm avoid local optima and potentially find the global optimum or high-quality solutions[22].

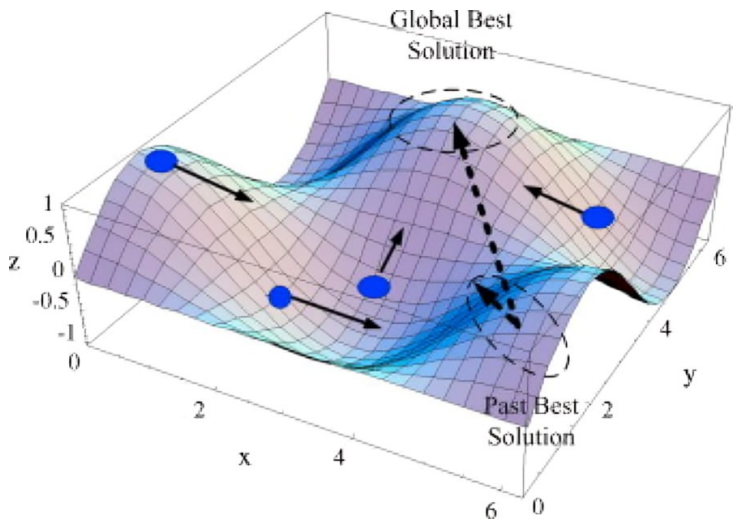


Figure 1.5: 3D example of particle swarm optimization exploring solution space to find optimal solutions.

[23]

1.3.4.2 Key Characteristics

- **Swarm Intelligence**: Uses collective behavior to explore the search space.
- **Velocity and Position Update**: Particles adjust their positions based on individual and group knowledge.

1.3.4.3 Applications

- **Optimization**: Solving continuous optimization problems in various domains.
- **Control Systems**: Tuning parameters in control systems for better performance.

Example: In logistics, PSO can optimize delivery routes by having particles represent possible routes and adjust based on delivery times and traffic patterns, aiming to minimize travel time and cost[24].

1.3.5 Ant Colony Optimization (ACO)

1.3.5.1 Principle

Inspired by the foraging behavior of real ants, Ant Colony Optimization (ACO) addresses complex optimization problems by finding optimal paths or solutions in scenarios with intricate connections and multiple potential options. The algorithm simulates a colony of artificial ants

navigating the search space, depositing a virtual substance called pheromone on the paths they travel. The amount of pheromone reflects the quality of a path based on a specific problem definition, such as shorter distance or lower cost.

Each ant probabilistically selects its next move based on the existing pheromone trail and possibly some heuristic information. Paths with stronger pheromone trails are more likely to be chosen, which encourages other ants to follow. Over time, the pheromone trails gradually evaporate, mimicking the natural decay of real pheromones. This evaporation prevents stagnation and promotes the exploration of new paths, potentially leading to the discovery of better solutions.

After their exploration, successful ants deposit additional pheromone on the paths they traveled, reinforcing the trails on good paths and guiding subsequent ants toward better solutions. This process of ant movement, pheromone evaporation, and pheromone update repeats over many cycles. Through this collaborative exploration and pheromone-based communication, the colony of artificial ants progressively discovers and converges on good solutions within the search space, effectively navigating complex landscapes and potentially finding the global optimum or high-quality solutions[25].

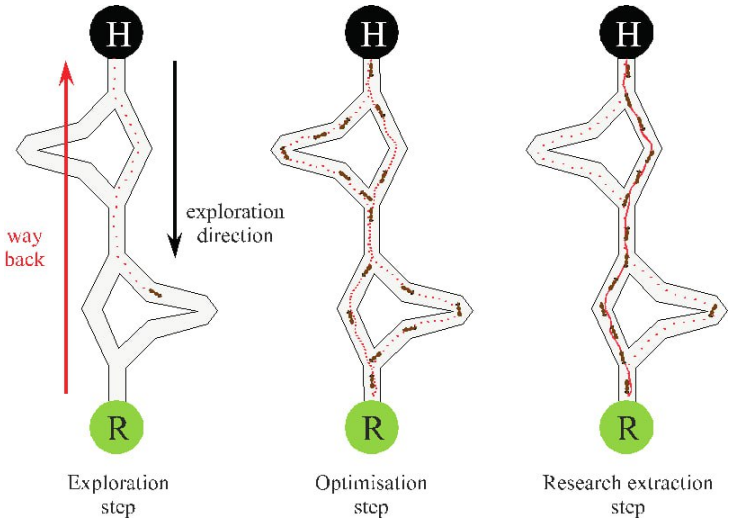


Figure 1.6: Ant colony exploring Solution space to find optimal solutions.
[26]

1.3.5.2 Key Characteristics

- **Pheromone Trails**: Paths with more pheromones are more likely to be followed.
- **Positive Feedback**: Successful paths attract more ants, reinforcing good solutions.

1.3.5.3 Applications

- **Network Routing**: Finding optimal paths in communication networks.
- **Scheduling**: Optimizing task sequences in manufacturing.

Example: In communication networks, ACO can optimize routing by having artificial ants explore various paths and reinforce the most efficient routes with pheromones, leading to improved data flow and reduced congestion.

Mastering and applying metaheuristic techniques enables researchers and practitioners in AI to address complex optimization challenges across a wide range of fields, from operations research to bioinformatics. The efficiency and adaptability of these techniques in exploring vast solution spaces and finding near-optimal solutions in diverse and intricate environments underscore their pivotal role in advancing both theoretical knowledge and practical applications[27].

1.4 Machine learning techniques

1.4.1 Definition

Machine learning, a subfield of artificial intelligence, focuses on developing and studying algorithms that can learn from data without explicit programming. Unlike traditional programming, where instructions are meticulously defined, machine learning algorithms iteratively improve their performance on specific tasks through exposure to data.

Machine learning algorithms rely on data to learn and improve, which can be labeled (where the desired outcome is known) or unlabeled (where the algorithm identifies patterns itself). The quality and quantity of data significantly impact the learning process and the effectiveness of the resulting model. Algorithmic learning models are mathematical frameworks that capture the relationships and patterns within the data, including decision trees, neural networks, and support vector machines. Each model type is suited for different problems and data types.

The learning process involves training the model on a portion of the data, during which the model adjusts its internal parameters to minimize errors in its predictions or classifications. The model's performance is then evaluated on a separate hold-out set of data to assess its generalization ability, meaning its capacity to perform well on unseen data. A key objective in machine learning is to train models that can generalize well to new, unseen data, ensuring the learned

patterns are not overly specific to the training data and can be effectively applied to real-world scenarios.

Machine learning has applications in diverse fields such as computer vision (image recognition, object detection), natural language processing (machine translation, sentiment analysis), recommender systems (product recommendations, personalized content), fraud detection (identifying suspicious financial activity), and scientific discovery (analyzing large datasets for patterns and insights). By leveraging data and algorithms, machine learning enables computers to learn and adapt without explicit programming, leading to significant advancements in various scientific and technological domains[28].

1.4.2 Reinforcement learning

1.4.2.1 Principle

Reinforcement learning (RL) is a subfield of machine learning focused on training an agent to make optimal decisions in an environment through trial and error. Unlike supervised learning, where the desired outcome is explicitly provided, RL deals with scenarios where the desired behavior is not explicitly defined but can be learned through interaction with the environment. At its core, RL involves an agent interacting with an environment, perceiving it through sensors (observations), and taking actions that modify the environment's state. The environment provides feedback to the agent in the form of rewards, which can be positive (reinforcement) for actions that move the agent closer to the desired goal, or negative (penalties) for actions leading it away[29].

The agent's objective is to learn a policy—a mapping from observations to actions—that maximizes its long-term reward. This is achieved through trial and error, exploring different actions and learning from the resulting rewards. Key elements of reinforcement learning include the state space (all possible states the environment can be in), the action space (all possible actions the agent can take), the reward function (defines the reward an agent receives for specific actions in particular states), the value function (estimates the expected future reward from a given state, considering the policy), and the policy function (maps observations to actions).

There are various approaches to learning an optimal policy in RL, including value-based methods (which estimate the value of each state or state-action pair to guide the agent towards higher-value actions), policy-based methods (which directly learn the policy function), and

model-based methods (which learn a model of the environment dynamics to simulate potential outcomes before taking actions, although this is less common due to the difficulty of accurately modeling complex environments)[30].

Reinforcement learning finds applications in diverse domains, such as training robots to navigate and manipulate objects effectively in robotics, enabling agents to learn and master complex games like Go or StarCraft II in game playing, optimizing resource allocation in complex systems for resource management, and personalizing recommendations based on user interactions in recommendation systems. By enabling agents to learn through interaction and reward signals, reinforcement learning offers a powerful framework for tackling problems where the desired behavior is not explicitly defined but can be learned through experience[31].

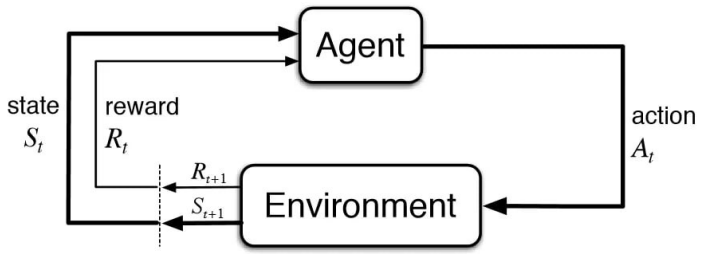


Figure 1.7: The iterative cycle of Reinforcement Learning.
[32]

1.4.2.2 Key Characteristics

- **Trial and Error Learning**: Agents learn through iterative interaction with the environment, exploring different actions and learning from received rewards.

- **Reward Optimization**: The primary goal is to maximize cumulative rewards over time, influencing the agent's decision-making process.

- **Temporal Credit Assignment**: Actions taken early in the sequence affect future rewards, requiring agents to attribute credit accurately.

1.4.2.3 Applications

- **Robotics**: Teaching robots to perform complex tasks like autonomous navigation or object manipulation.

- **Game Playing**: Training agents to excel in strategy games such as chess or video games like Dota 2[33].

1.4.3 Graph neural network (GNN)

1.4.3.1 Principle

Graph neural networks (GNNs) are a specific type of deep learning architecture designed to operate on graph-structured data. Unlike traditional neural networks that process vectors or matrices, GNNs excel at capturing the relationships and interactions between nodes (entities) and edges (connections) within a graph. This capability makes them well-suited for tasks on various graph-based problems across diverse domains. Here's a breakdown of the core principles of GNNs:

- **Graph Representation**: A graph is a mathematical structure consisting of nodes (vertices) representing entities and edges (links) representing connections between those entities. GNNs typically represent nodes with feature vectors containing relevant information about the entity, and edges might have additional attributes depending on the specific problem.

- **Message Passing**: The core principle of GNNs lies in the concept of message passing. During the training process, information is iteratively propagated through the graph. Each node aggregates information from its neighbors (nodes it's connected to) through edge-based messages. These messages can be computed using learnable functions that consider the features of both the sending node and the connecting edge.

- **Node Aggregation and Update**: After receiving messages from neighbors, a node aggregates the information using learnable aggregation functions. This aggregated information is then combined with the node's own features through a non-linear transformation (e.g., using activation functions) to update the node's representation. This updated representation incorporates the node's inherent features and the information received from its neighbors.

- **Multiple Layers**: Similar to traditional neural networks, GNNs can have multiple layers. In each layer, the message passing, aggregation, and update steps are repeated, allowing

the network to learn increasingly complex representations of nodes and their relationships within the graph structure.

- **Output Layer**: The final layer of a GNN typically depends on the specific task.

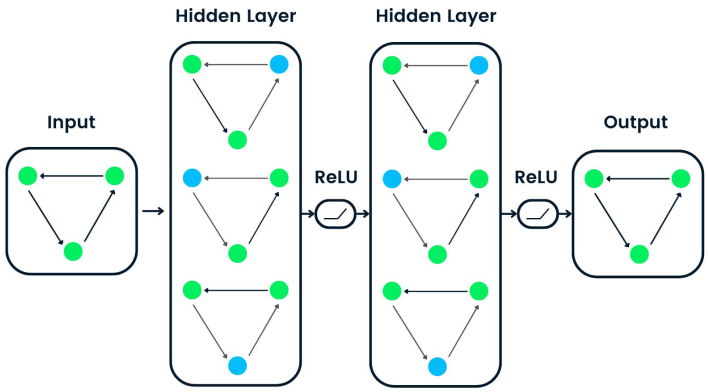


Figure 1.8: A visual guide to understand graph neural network architecture. [34]

For node-level prediction tasks (e.g., classifying a node as belonging to a particular category), the node representations from the final layer can be used for classification.
For graph-level prediction tasks (e.g., predicting properties of the entire graph), the node representations can be aggregated using various pooling functions to obtain a graph-level representation for prediction.
By leveraging message passing and iterative information aggregation, GNNs can effectively capture the inherent relational structure of graph data. This capability allows them to tackle various tasks, including:

- **Node classification**: Classifying nodes into different categories based on their features and connections.
- **Link prediction**: Predicting the likelihood of a link existing between two nodes.
- **Community detection**: Identifying groups of nodes within a graph that are densely connected.
- **Graph generation**: Learning to generate new graphs with similar properties to a given set of graphs.

GNNs represent a powerful tool for leveraging the rich information embedded within graph structures, leading to advancements in social network analysis, recommender systems, protein-protein interaction analysis in biology, and many other application areas[35].

1.4.3.2 Key characteristics

- **Relationship and Interaction Modeling**: GNNs excel at capturing the intricate relationships and interactions between nodes (entities) and edges (connections) within a graph, making them suitable for tasks that involve understanding and reasoning about complex relational data.

- **Message Passing**: The core principle of GNNs involves iteratively passing and aggregating information between connected nodes in a graph. This allows nodes to update their representations based on information from neighboring nodes, facilitating the learning of graph-level features.

- **Graph Representation Learning**: GNNs learn to represent nodes and edges with feature vectors that encapsulate relevant information about each entity and its relationships within the graph. These representations are crucial for downstream tasks such as node classification or graph-level predictions.

- **Multi-layer Architecture**: Similar to traditional neural networks, GNNs can consist of multiple layers where each layer iteratively refines node representations through message passing, aggregation, and transformation steps. This hierarchical approach enables the model to learn increasingly abstract and complex features.

1.4.3.3 Applications

- **Node Classification**: Classifying nodes into different categories based on their features and relationships within the graph. For example, identifying the topic of documents in a citation network.

- **Link Prediction**: Predicting the likelihood of a connection (edge) between nodes that are not currently linked in the graph. This is useful in recommender systems or social network analysis.

- **Graph Classification**: Classifying entire graphs into categories based on their structural properties. For instance, classifying molecular graphs based on their chemical properties.
- **Graph Generation**: Generating new graphs that exhibit similar structural characteristics as the input graphs. This is useful in generating molecules with specific properties or creating realistic social networks for simulation purposes.
- **Social Network Analysis**: Analyzing social connections, influence propagation, and community detection in social networks.
- **Recommender Systems**: Providing personalized recommendations based on user-item interactions represented as graphs.
- **Biological and Chemical Studies**: Analyzing molecular structures, protein-protein interactions, or drug discovery tasks where molecules are represented as graphs.
- **Traffic Flow Prediction**: Predicting traffic congestion and optimizing transportation networks where road intersections and traffic flow can be modeled as graphs.

GNNs represent a powerful tool for leveraging the rich relational information inherent in graph structures, enabling advancements in diverse fields from social sciences to computational biology and beyond[36].

1.5 Conclusion

This chapter focused on artificial intelligence techniques applied to optimization, highlighting a variety of approaches and their applications in different contexts. We began with the foundations of graph theory, emphasizing its crucial role in modeling and solving complex optimization problems. Graph traversal algorithms such as Breadth-First Search (BFS), Dijkstra's Algorithm, and A* Search Algorithm were examined in detail, illustrating their respective capabilities in finding optimal paths in networks, then explored metaheuristic techniques, such as Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization. These methods, inspired by natural and biological processes, have proven effective in solving difficult optimization problems by finding approximate but often highly efficient solutions. Machine learning techniques were discussed, highlighting their ability to adapt and learn from

data to improve optimization performance. In particular, reinforcement learning was examined for its application in dynamic environments where decisions must be made sequentially. Finally, Graph Neural Networks (GNN) were presented as a promising advancement, combining the advantages of graphs and neural networks to solve complex optimization problems in non-Euclidean data structures. In conclusion, this chapter has demonstrated that optimization is a multidisciplinary field where artificial intelligence techniques play an essential role. Each method discussed offers unique advantages and can be adapted to the specificities of the problems encountered. The synergy between these techniques allows for the development of robust and innovative solutions, paving the way for new applications and continuous improvements in the field of optimization. In the next chapter, we will focus on the optimization of public transportation systems. We will analyze how artificial intelligence techniques can be applied to improve the efficiency and quality of public transportation services, taking into account the specific constraints and requirements of this sector.

Chapter 2

AI Techniques In Public Transportation Optimization

2.1 Introduction

Urban mobility networks serve as the circulatory system of cities, enabling vital movement and economic vitality. As urban areas expand and travel patterns shift, these networks face mounting pressure to deliver seamless, dependable, and inclusive services. Consequently, the optimization of urban transit becomes paramount. This field blends engineering concepts, mathematical frameworks, and data-driven insights to boost the efficacy of public transport systems. Urban transit encompasses a range of modalities, from buses and trains to trams, metros, and water-based options, operating within an intricate web of pathways, access points, and timetables. Grasping these components is essential for enhancing performance. Pathways represent the designated routes vehicles traverse to link starting points and destinations. Access points are specific locations where passengers embark or disembark. Timetables outline the projected arrival and departure times at each access point. The success of an urban transit system hinges on the harmonious interplay of these elements: streamlined pathways reduce journey durations, strategically positioned access points enhance reach, and well-synchronized timetables ensure smooth passenger flows. The significance of enhancing urban transit lies in tackling several critical issues. Streamlining involves curtailing travel times, waiting periods, and overall trip lengths for users. Dependability ensures punctual service and minimizes disruptions. Inclusivity provides accessible and cost-effective transit options for all demographic groups. Eco-consciousness aims to reduce environmental impact by maximizing passenger capacity

and decreasing reliance on personal vehicles. By employing enhancement techniques, transit authorities can achieve these objectives, resulting in a more appealing and sustainable mobility ecosystem. Cutting-edge innovations play an increasingly vital role in bolstering urban transit enhancements. Real-time data from positioning systems, sensors, and user applications enable dynamic adjustments to pathways and timetables based on current traffic conditions and passenger demand. Virtual modeling tools allow planners to simulate various scenarios and evaluate the impact of potential changes before implementation. Machine learning algorithms can process vast datasets and optimize complex decision-making processes, such as pathway planning and vehicle deployment. Integrating these innovations into urban transit systems can foster a more efficient, responsive, and user-centric experience for passengers.

2.1.1 Congestion and Overcrowding

- **Urban Traffic**: Public transportation systems often operate within congested road networks, leading to delays and reduced travel speeds. This discourages ridership and undermines the system's efficiency. Traffic congestion not only delays buses but also impacts the timeliness of interconnected services such as feeder buses and trams, creating a ripple effect throughout the network.

- **Capacity Limitations**: Buses and trains can reach maximum capacity during peak hours, resulting in overcrowding and discomfort for passengers. This overcrowding can deter potential riders and create safety concerns. The discomfort and safety issues associated with overcrowding can significantly diminish the attractiveness of public transportation, pushing commuters towards private vehicles.

2.1.2 Scheduling and Timeliness Issues

- **Unreliable Schedules**: Delays caused by unexpected events, such as mechanical breakdowns, accidents, or inclement weather, can disrupt schedules and frustrate passengers. Unreliable schedules lead to reduced ridership and decreased trust in the system. Passengers may resort to alternative transportation modes if they consistently experience unpredictable service.

- **Frequency and Headways**: Infrequent service, particularly during off-peak hours, can result in long wait times and discourage ridership. Passengers with limited time or inflex-

ible schedules may find the service inconvenient, reducing the overall appeal of public transportation. Ensuring consistent and frequent service across all hours of operation is essential to maintaining a reliable and user-friendly system.

2.1.3 Accessibility and Inclusivity Concerns

- **Physical Barriers**: Physical barriers, such as stairs with no ramps or lack of elevators at stations, can make public transportation inaccessible for people with disabilities, the elderly, or those with young children using strollers. Improving physical accessibility is crucial to ensuring that all community members can use public transportation.

- **Social and Economic Barriers**: High fares, complex ticketing systems, and lack of information in multiple languages create economic and social barriers for low-income residents and non-native speakers, hindering inclusivity. Simplifying ticketing processes and providing multilingual information can significantly enhance accessibility for diverse populations.

2.1.4 Environmental Impact

While public transportation typically has a lower environmental impact per passenger compared to private vehicles, it still contributes to air pollution through vehicle emissions, especially from older diesel buses. Additionally, energy consumption for electric trains and subways can have environmental implications if the electricity is generated from non-renewable sources. Adopting greener technologies, such as electric or hybrid buses and sourcing renewable energy for electric trains, can mitigate these environmental impacts[37].

2.2 Interconnected Challenges and Comprehensive Solutions

These challenges are interconnected and can amplify each other. For instance, congestion can lead to schedule delays, further discouraging ridership and increasing reliance on private vehicles, which exacerbates congestion. Addressing these challenges requires a comprehensive approach that combines infrastructure improvements, technological advancements, and strategic planning. By tackling these issues holistically, public transportation can be transformed into a more efficient, reliable, equitable, and sustainable mode of travel.

Infrastructure improvements might include expanding road networks dedicated to buses, adding more train cars, or upgrading stations for better accessibility. Technological advancements such as real-time tracking, predictive maintenance, and automated scheduling can enhance reliability and efficiency. Strategic planning involves coordinating transportation policies, land use planning, and community engagement to ensure that public transportation systems meet the needs of all residents. Through these concerted efforts, public transportation can become a cornerstone of urban mobility, fostering economic growth, social inclusion, and environmental sustainability[38].

2.2.1 Challenges in Developing digital Tools for Transportation Optimization

In addition to the operational challenges faced by public transportation systems, developing effective tools for transportation optimization presents its own set of difficulties. Addressing these technical and logistical challenges is critical to harnessing the full potential of technology in public transportation.

2.2.1.1 Data Security and Privacy

- **Securing Data**: Ensuring that stored and communicated data is protected against unauthorized access, whether through disclosure or alteration, is paramount. This requires robust cryptography and access control measures tailored to different types of data interactions.

- **Data Storage**: Storing vast amounts of collected data while preserving privacy and integrity is a significant challenge. Strategies must be developed to manage data in ways that protect sensitive information from breaches and misuse.

2.2.1.2 Data Collection and Quality

- **Types of Data**: Determining what types of data can be collected, both in real-time and retrospectively, is essential for effective transportation optimization. This includes data on passenger numbers, vehicle locations, traffic conditions, and more.

- **Data Quality**: Ensuring the collected data is of sufficient quality is crucial for accurate

analysis and decision-making. Poor quality data can lead to incorrect conclusions and suboptimal optimization strategies.

- **Data Quantity**: Handling situations with insufficient data requires developing methods to fill gaps and make reliable inferences, ensuring continuous and effective optimization even when data is sparse.

2.2.1.3 Data Presentation and Interaction

- **User-Friendly Data Presentation**: Presenting data and information in ways that are easily understandable and usable by both humans and machines is critical. Effective visualization and interface design can significantly enhance the usability of transportation optimization tools.

- **Intuitive Interaction Models**: Designing appropriate interaction models so that users can intuitively interact with services and devices is necessary for widespread adoption. This includes user-friendly applications and interfaces that facilitate seamless engagement with the system.

2.2.1.4 Data Analysis and Processing

- **Transforming Data into Information**: Analyzing and processing the vast amounts of collected data to convert it into meaningful information is a complex task. This involves employing advanced analytics, machine learning, and artificial intelligence to derive actionable insights that can inform optimization strategies[39].

2.2.2 Comprehensive Approach to Addressing Challenges

To overcome these challenges, a comprehensive approach is required, integrating robust security protocols, advanced data management techniques, user-centric design, and sophisticated data analysis methods. By addressing both operational and developmental challenges, public transportation systems can be optimized to deliver efficient, reliable, and user-friendly services. This holistic strategy will enable cities to create more sustainable and inclusive transportation networks, meeting the diverse needs of their populations.

2.3 Role of Artificial Intelligence in Transportation

Artificial intelligence (AI) is rapidly transforming the transportation landscape, offering significant potential to address many of the challenges faced by public transportation systems. This transformation is evident in various applications, from demand prediction and route optimization to traffic management and predictive maintenance, all contributing to the overarching framework of smart cities.

2.3.1 Demand Prediction and Route Optimization

AI algorithms can analyze historical data on ridership, traffic patterns, and weather conditions to predict passenger demand in real time. This capability allows for dynamic route adjustments, directing vehicles to areas with high passenger volume while avoiding congested areas. Furthermore, improved scheduling becomes possible, optimizing schedules to align with predicted demand, thereby reducing wait times and overcrowding during peak hours and increasing service frequency in areas with high demand.

2.3.2 Enhanced Traffic Management

AI-powered traffic management systems analyze real-time traffic data to optimize traffic light timing based on traffic flow, reducing congestion and improving overall network efficiency. These systems can also predict and reroute traffic by forecasting incidents and congestion hotpots, allowing for the rerouting of vehicles and providing real-time traffic updates to passengers.

2.3.3 Predictive Maintenance and System Reliability

AI can analyze sensor data from vehicles and infrastructure to predict potential equipment failures. This enables preventative maintenance, allowing for early detection of potential issues and proactive maintenance actions, which minimizes disruptions and improves system reliability. By addressing issues before they cause breakdowns, AI ensures smoother operation of public transportation systems, reducing downtime and enhancing overall efficiency[40].

2.3.4 Improved Passenger Experience

AI-powered chatbots can provide real-time information to passengers regarding schedules, delays, and route changes. Personalized route recommendations can be offered based on individual passenger needs and preferences, while AI-powered facial recognition systems can streamline ticketing and boarding processes, enhancing the overall passenger experience.

2.3.5 Intelligent Transportation Systems (ITS)

AI technologies are used in ITS to monitor and manage transportation networks. This includes real-time incident detection, dynamic routing, adaptive traffic signal control, and providing traveler information to enhance safety and efficiency.

2.3.6 Passenger Experience

AI-powered virtual assistants and chatbots enhance the passenger experience by providing real-time information, personalized recommendations, and assistance throughout the journey.

The integration of AI in transportation systems has the potential to revolutionize travel, making it safer, more efficient, and sustainable. However, addressing ethical considerations, cybersecurity, and the impact on employment is essential to ensure the responsible and beneficial implementation of AI in transportation.[41].

2.4 Deep Reinforcement Learning and Graph Neural Networks in optimization of public transportation

Public transportation systems face a complex challenge of optimizing routes, schedules, and resource allocation while navigating real-time traffic, passenger demand fluctuations, and network constraints. Traditional methods often struggle to adapt to the dynamic nature of these environments. This is where the synergy between Deep Reinforcement Learning (DRL) and Graph Neural Networks (GNNs) emerges as a potent solution for transportation optimization.

2.4.1 Deep Reinforcement Learning in transportation

Deep reinforcement learning (DRL) has emerged as a promising approach to address complex optimization problems within transportation systems. Unlike traditional methods, DRL integrates machine learning techniques with reinforcement learning principles to enable agents to learn optimal policies through interaction with their environment. In transportation, this methodology proves particularly effective in dynamically adjusting routing strategies, scheduling protocols, and traffic management decisions based on real-time data inputs. One key application of DRL in transportation involves route optimization, where agents learn to adaptively adjust routes in response to changing traffic conditions and passenger demand. By continuously interacting with the environment and receiving feedback through rewards and penalties, DRL agents can optimize travel times and resource utilization, ultimately enhancing system efficiency and user satisfaction.

Furthermore, DRL facilitates adaptive traffic signal control, allowing for real-time adjustments in signal timings based on current traffic flow patterns. This capability not only improves traffic flow but also reduces congestion and enhances overall network performance, making it a valuable tool for urban mobility management.

Despite its potential, the application of DRL in transportation also presents challenges such as computational complexity, data requirements, and the need for robust decision-making frameworks. Addressing these challenges requires ongoing research and development to refine algorithms, improve scalability, and ensure the reliability and safety of DRL-based transportation systems.

In conclusion, deep reinforcement learning offers a transformative approach to optimizing transportation systems, leveraging real-time data and adaptive decision-making to improve efficiency, sustainability, and user experience. Continued advancements in DRL methodologies hold promise for revolutionizing urban mobility and shaping the future of intelligent transportation systems[42].

2.4.2 Graph Neural Network and optimization

Graph neural networks (GNNs) have emerged as powerful tools for analyzing and optimizing transportation networks. Unlike traditional neural networks, GNNs are designed to handle data structured as graphs, where nodes represent entities (such as intersections or stations) and edges

represent relationships (such as roads or transit lines).

In transportation applications, GNNs excel at capturing the complex relationships and dependencies within networks. They can effectively model the flow of traffic, predict congestion patterns, and optimize routing strategies based on real-time data inputs. By leveraging the spatial and temporal dependencies inherent in transportation networks, GNNs enable more accurate and efficient decision-making processes.

One significant application of GNNs in transportation is in traffic prediction and management. By integrating real-time traffic data and historical patterns, GNNs can forecast traffic conditions and recommend adaptive strategies such as dynamic route adjustments or signal control optimizations. This capability not only improves travel efficiency but also enhances safety and reduces environmental impact by minimizing congestion and emissions.

Furthermore, GNNs are valuable in urban mobility planning and optimization. They can analyze the connectivity and accessibility of transit networks, identify critical nodes or links, and propose infrastructure improvements to enhance overall network performance. This holistic approach helps city planners and transportation authorities make informed decisions for improving public transit systems and accommodating growing urban populations.

Despite their effectiveness, deploying GNNs in transportation systems poses challenges such as scalability, interpretability of results, and integration with existing infrastructure. Addressing these challenges requires ongoing research efforts to refine GNN architectures, develop efficient training algorithms, and ensure compatibility with diverse transportation datasets.

In conclusion, graph neural networks represent a transformative technology for enhancing transportation networks' efficiency, resilience, and sustainability. By harnessing the power of graph-based modeling, GNNs offer innovative solutions to complex transportation challenges and pave the way for smarter and more adaptive urban mobility systems[43].

2.4.3 DRL and GNNs

The integration of Deep Reinforcement Learning (DRL) with Graph Neural Networks (GNNs) leverages their respective strengths to create a symbiotic relationship. The DRL agent utilizes the GNN's insights to make informed decisions on route planning, scheduling adjustments, and traffic signal control, thereby optimizing overall transportation efficiency. Conversely, GNNs benefit from DRL's adaptive learning capabilities, facilitating the development of robust and flexible optimization strategies that can adapt to changing network dynamics and operational

conditions. This collaboration enhances decision-making processes within transportation systems, leading to more efficient resource allocation and improved system performance.
Examples:

- **Dynamic Bus Routing**: An integrated DRL-GNN system can dynamically suggest different buses line in response to real-time traffic congestion and passenger demand fluctuations, enhancing service efficiency and passenger satisfaction.
- **Demand-Responsive Transit**: Using DRL and GNNs, on-demand transit systems can be created where vehicle dispatches align closely with real-time passenger requests, offering flexible and responsive public transportation solutions.

2.4.3.1 Challenges and Considerations

- **Computational Demands**: Implementing DRL and GNNs requires substantial computational resources, necessitating high-performance computing environments for effective deployment.
- **Explainable AI**: Ensuring transparency and interpretability of decision-making processes remains a critical area of research, particularly in addressing biases and building public trust in automated transportation systems[44].

DRL integrated with GNNs presents a robust framework for optimizing public transportation systems, promising enhanced efficiency, reliability, and sustainability. As advancements continue in explainable AI and computational efficiency, these intelligent transportation systems are poised to transform urban mobility, paving the way for smarter, more responsive, and user-centric transportation networks of the future.

2.5 Conclusion

The optimization of public transportation is an essential endeavor in modern urban planning, aiming to enhance efficiency, reliability, equity, and sustainability. As cities continue to grow and the demand for effective transit solutions increases, leveraging advanced technologies and innovative methodologies becomes paramount. Throughout this chapter, we have explored the multifaceted challenges and opportunities within public transportation systems, emphasizing the pivotal role of optimization techniques and technological advancements.

The key challenges identified, such as congestion, overcrowding, scheduling issues, accessibility concerns, and environmental impacts, require a comprehensive approach to develop effective solutions. Addressing these issues involves infrastructure improvements, strategic planning, and the integration of cutting-edge technologies like artificial intelligence, deep reinforcement learning, and graph neural networks. These advancements enable dynamic adjustments, predictive maintenance, and personalized passenger experiences, ultimately leading to more responsive and efficient public transportation systems.

Artificial intelligence, particularly in the forms of demand prediction, route optimization, and enhanced traffic management, plays a transformative role in modernizing public transportation. By analyzing vast amounts of data in real-time, AI can optimize schedules, reduce congestion, and improve overall system reliability. The incorporation of deep reinforcement learning and graph neural networks further strengthens these capabilities, allowing for adaptive decision-making and efficient resource allocation.

Despite the significant progress, the development and deployment of these technologies present their own set of challenges, including data security, computational demands, and the need for explainable AI. Addressing these challenges requires ongoing research and innovation to ensure the safe and effective implementation of advanced optimization tools in public transportation.

In conclusion, optimizing public transportation systems is a complex but critical task that necessitates a holistic approach. By embracing technological advancements and innovative methodologies, cities can create sustainable, equitable, and efficient transit networks. These improvements will not only enhance the passenger experience but also contribute to broader societal goals of economic growth, social inclusion, and environmental sustainability. The future of public transportation lies in the successful integration of these advanced optimization techniques, paving the way for smarter and more adaptive urban mobility solutions.

Chapter 3

Conception

3.1 Introduction

In this chapter, we delve into the conception of a transformative application designed to revolutionize the public transportation experience. The primary objective of this project is to create an intuitive and efficient solution that empowers users to navigate public transit systems independently, without the need to seek assistance from others. By leveraging cutting-edge artificial intelligence technologies, this application aims to optimize the utilization of public transportation and simplify the commuting process for all users. Public transportation networks, while essential for urban mobility, often present challenges in terms of route planning, schedule adherence, and overall convenience. Our proposed application addresses these challenges by providing accurate, real-time information and personalized guidance, tailored to each traveler's specific needs and preferences. This chapter outlines the conceptual framework of the project, highlighting the innovative use of AI to enhance user experience and promote greater efficiency in public transportation systems. We will explore the core components and functionalities of the application, detailing how AI algorithms can process vast amounts of transportation data to deliver precise, user-centric solutions. Additionally, the chapter will discuss the anticipated benefits of the application, including reduced wait times, optimized travel routes, and increased accessibility for all users, particularly those unfamiliar with the intricacies of public transit systems. Through this conceptual overview, we aim to provide a comprehensive understanding of the project's vision and the technological advancements that will drive its success.

3.2 The goal of our application

The goal of our application is to enhance the public transportation experience by leveraging artificial intelligence to provide users with accurate, reliable, and personalized guidance. This will optimize the use of public transportation systems and simplify the commuting process, ensuring that users can navigate transit networks independently and efficiently without the need for external assistance. The application aims to reduce wait times, improve route planning, and increase overall convenience, making public transportation more accessible and user-friendly for all travelers.

3.3 Identification of actors

In the context of our public transportation optimization application, identifying the key actors is crucial for understanding the various roles and interactions within the system. The following is the only actor involved:

- **Users (travelers)**: Individuals who use the application to plan and navigate their public transportation routes are responsible for entering their travel preferences and destinations. They follow the guidance provided by the application and offer feedback on their experience.

3.4 Requirement specification

3.4.1 Functional Requirements

- **Authentication and Role Management**: The application should provide secure authentication mechanisms and manage different user roles, ensuring that each user has appropriate access rights and privileges.

- **Route Planning and Optimization**: Users should be able to plan their routes using the application, with AI-powered optimization to suggest the most efficient and time-saving routes based on real-time data.

- **Real-time Tracking and Updates**: The application should offer real-time tracking of public transportation vehicles, providing users with live updates on arrival times, delays,

and route changes.

- **Personalized Guidance and Recommendations**: The application should offer personalized guidance based on the user's preferences, travel history, and real-time conditions. This includes recommending the best transportation options, routes, and schedules.

- **Fare Calculation and Payment Integration**: The application should calculate fares for different routes and integrate with various payment systems to allow users to purchase tickets and passes directly through the app.

- **Accessibility Features**: The application must include features that make it accessible to users with disabilities, such as voice commands, text-to-speech, and high-contrast interfaces.

- **Feedback and Reporting Mechanism**: Users should be able to provide feedback on their travel experience and report issues directly through the app, facilitating continuous improvement of the service.

3.4.2 Non-functional Requirements

- **Security**: The application must be highly secure, ensuring that data is protected from unauthorized access. User data should only be accessible through secure authentication methods, such as user IDs and passwords. Additionally, encryption should be used to protect sensitive information both in transit and at rest.

- **Usability**: The application should be user-friendly, providing an intuitive and logical sequence of interfaces that make it easy for users to navigate and utilize all features effectively. This includes a clean and straightforward design, clear instructions, and responsive support for user queries.

- **Scalability**: The application should be scalable, meaning it should be able to handle an increasing number of users and data without compromising performance. It should support the addition of new features and services without requiring significant changes to the existing system.

- **Performance**: The application must be highly performant, with minimal response times to user actions. It should be capable of handling real-time data processing and updates

efficiently, ensuring a smooth user experience even during peak usage times.

- **Reliability**: The application should be reliable, ensuring that it is available and operational 24/7 with minimal downtime. Robust error handling and recovery mechanisms should be in place to maintain service continuity.
- **Maintainability**: The application should be maintainable, allowing developers to easily update and manage the codebase. This includes using clear and well-documented code, modular design principles, and automated testing.
- **Privacy**: The application must ensure the privacy of user data, adhering to relevant data protection regulations and standards. Users should have control over their data, including the ability to view, edit, and delete their personal information.

3.5 UML diagrams

Unified Modeling Language (UML) diagrams are standardized visual representations used in software engineering and system design. They depict the structure, behavior, and interactions of software systems, facilitating communication among stakeholders such as developers, designers, and project managers. UML diagrams include structural diagrams (e.g., class diagrams, object diagrams) that show static system elements, and behavioral diagrams (e.g., use case diagrams, sequence diagrams) that illustrate dynamic system behavior. These diagrams play a crucial role in analyzing, designing, and documenting software systems, ensuring clarity and consistency in system understanding and development processes.

3.5.1 Use case diagram

A use case diagram is a graphical representation within the Unified Modeling Language (UML) that illustrates the interactions between users (actors) and a system. It provides a high-level overview of the functionalities or behaviors that the system offers to its users, focusing on the external actors and their interactions with the system. Use case diagrams typically include actors (representing users or external systems), use cases (representing specific functionalities or tasks), and relationships (such as associations and dependencies) between them. These diagrams serve to capture and communicate the system's requirements from a user's perspective, aiding in the analysis, design, and validation phases of software development projects.

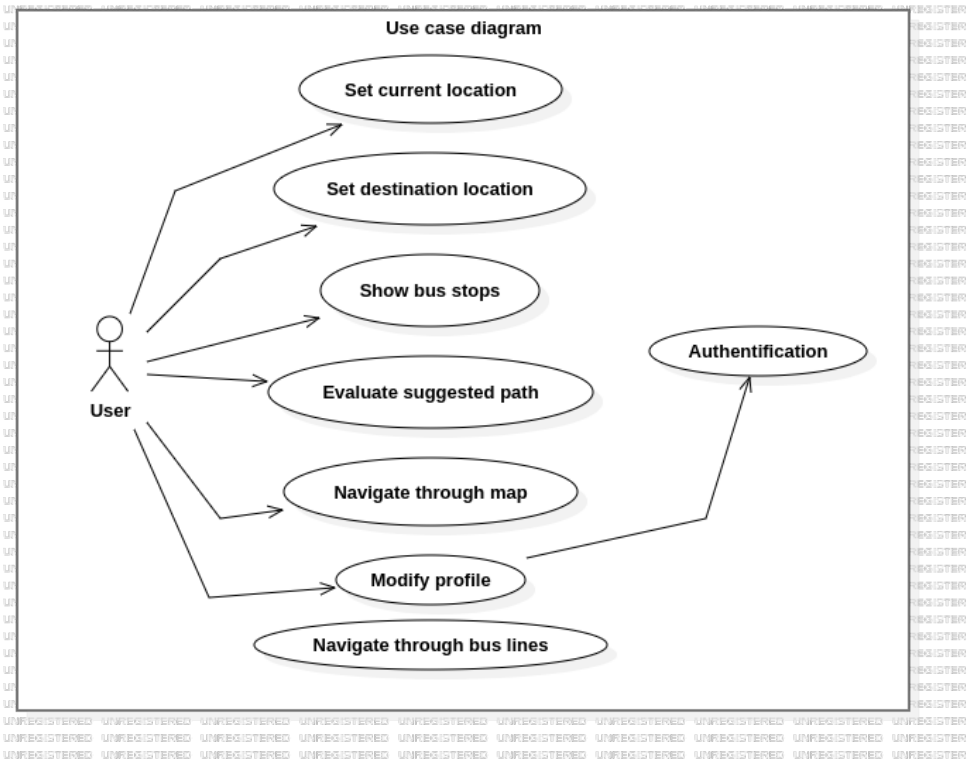


Figure 3.1: Use case diagram

3.5.2 Sequence diagram

A sequence diagram is a type of Unified Modeling Language (UML) diagram used in software engineering to depict interactions between objects or components within a system over time. It illustrates the sequence of messages exchanged between objects or roles (represented as life-lines) in a chronological order, showing how these interactions lead to the accomplishment of a specific functionality or scenario. Sequence diagrams emphasize the temporal aspect of system behavior, detailing the flow of control and data among objects in response to events or triggers. They are valuable for visualizing and analyzing the dynamic behavior of a system, aiding in the understanding, design, and implementation of software systems.

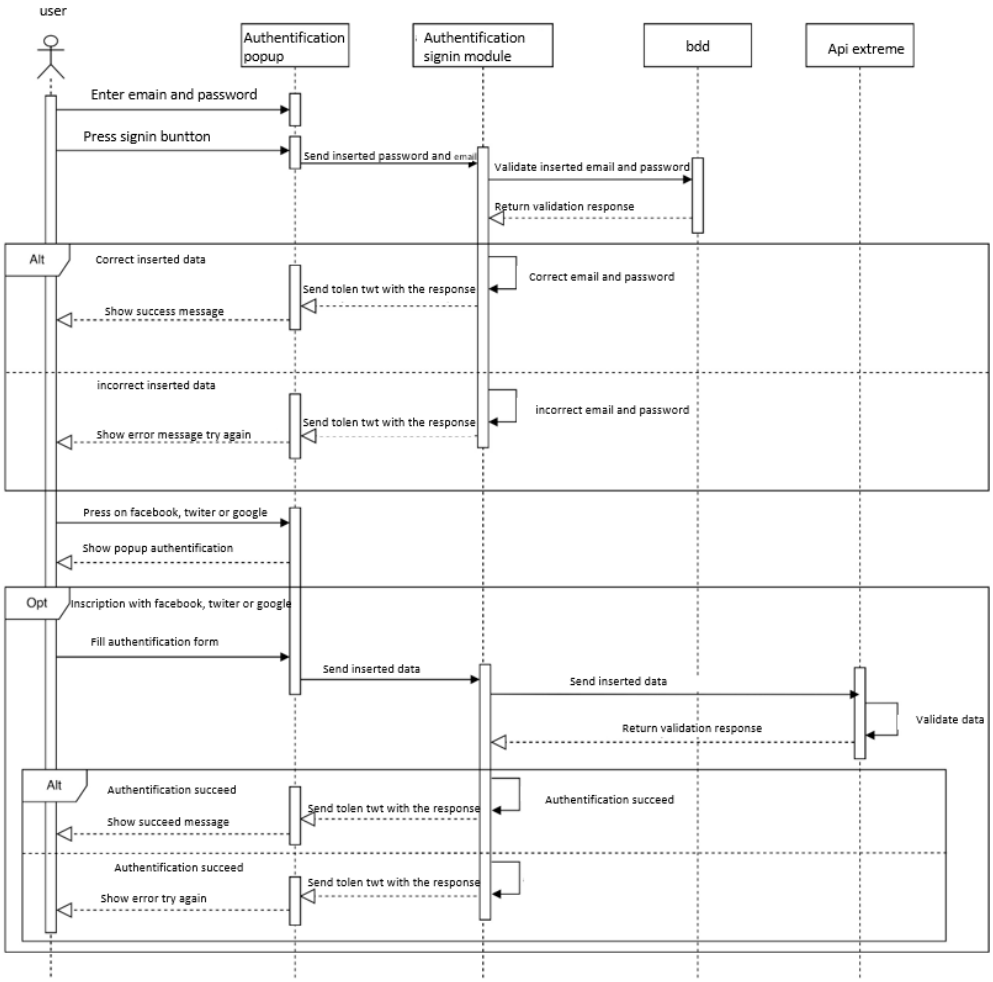


Figure 3.2: Sign in sequence diagram

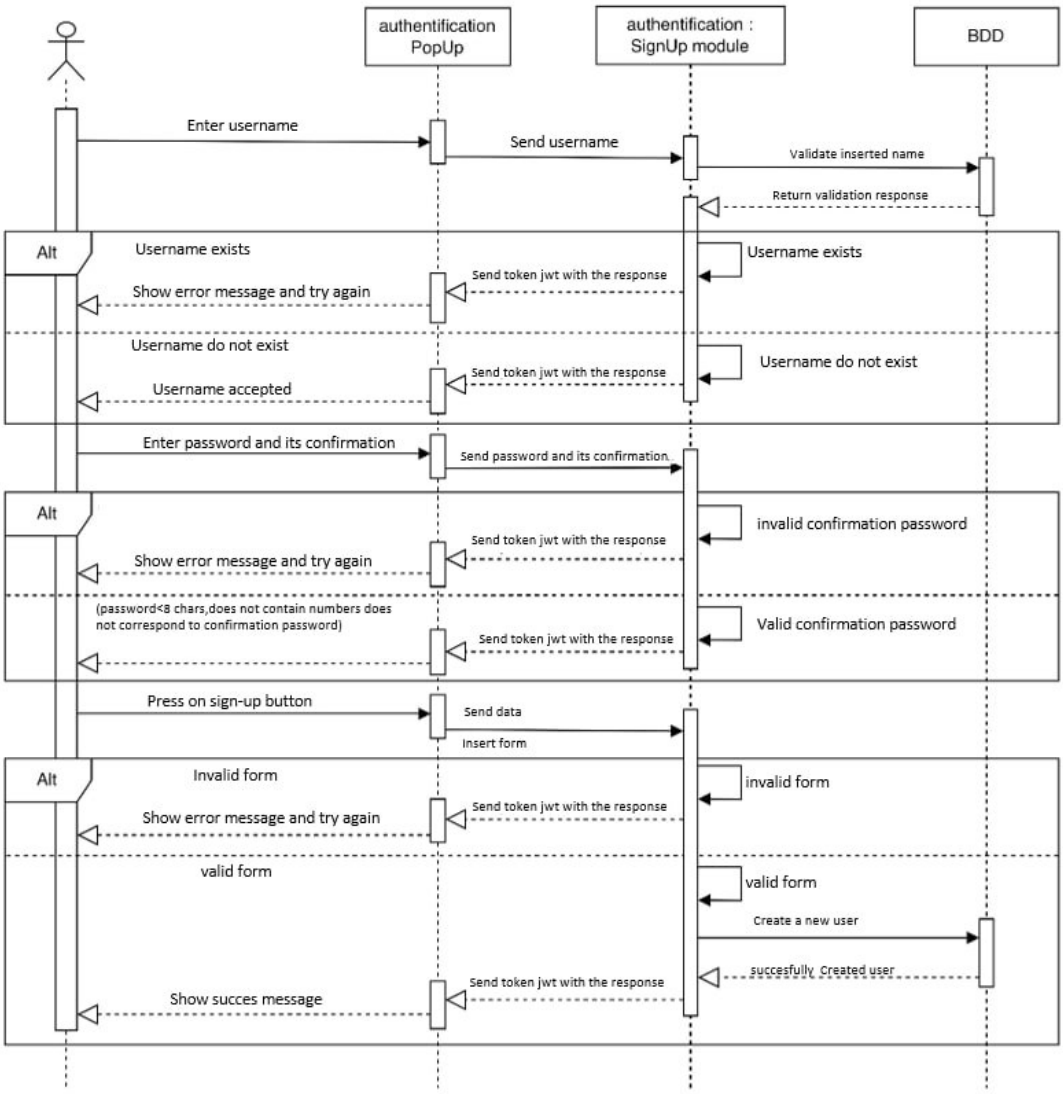


Figure 3.3: Sign up sequence diagram

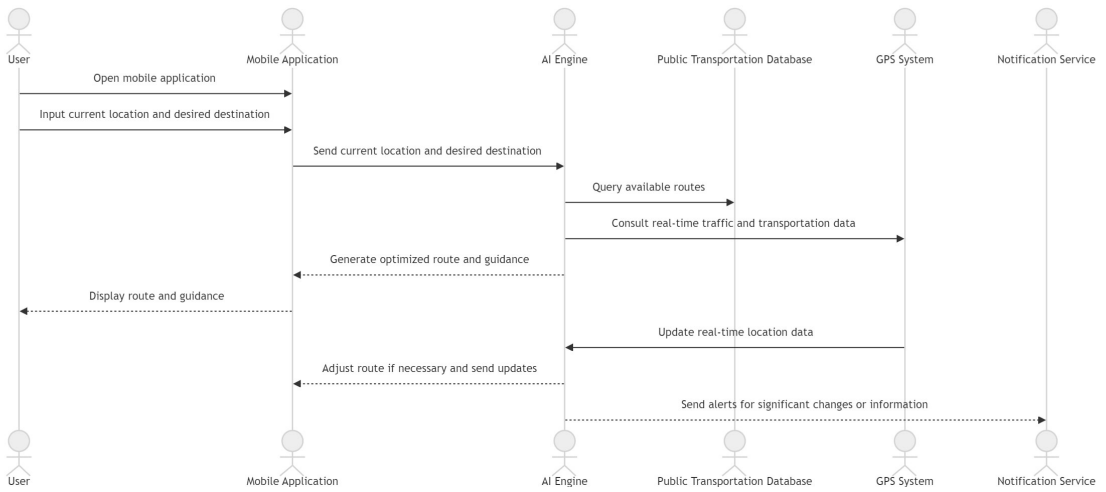


Figure 3.4: Navigation sequence diagram

3.6 Conclusion

In conclusion, the conceptual framework of our transformative public transportation application highlights the potential of leveraging artificial intelligence to revolutionize the commuting experience. This project aims to address the inherent challenges in public transportation networks, such as route planning, schedule adherence, and overall convenience, by providing accurate, real-time information and personalized guidance tailored to each user's needs and preferences.

The primary goal of our application is to enhance the public transportation experience, making it more accessible, reliable, and user-friendly for all travelers. By offering features such as AI-powered route planning, real-time tracking, personalized recommendations, and seamless fare payment integration, the application seeks to optimize the use of public transportation systems and simplify the commuting process. Additionally, the inclusion of accessibility features ensures that the application is usable by individuals with disabilities, promoting greater inclusivity.

Identifying the key actors, particularly the users, has allowed us to focus on their roles and interactions within the system, ensuring that the application meets their needs effectively. The functional and non-functional requirements specified for the application emphasize the importance of security, usability, scalability, performance, reliability, maintainability, and privacy. These requirements guide the development process, ensuring that the application not only meets users' expectations but also adheres to industry standards and best practices.

Through this conceptual overview, we have provided a comprehensive understanding of the

project's vision and the technological advancements that will drive its success. The anticipated benefits of the application include reduced wait times, optimized travel routes, and increased accessibility, ultimately enhancing the overall efficiency and user experience of public transportation systems. By harnessing the power of artificial intelligence, our application aims to create a more efficient, responsive, and user-centric public transportation network, paving the way for smarter urban mobility solutions.

Chapter 4

Work envirenment and development Tools

4.1 Introduction

In this chapter, we delve into the implementation phase of our public transportation optimization application. This section provides a comprehensive overview of the tools and technologies employed to bring the conceptual framework outlined in the previous chapter to life.

We will explore the various software and hardware components that underpin the application's functionality, including the artificial intelligence algorithms, data processing techniques, and user interface design. Additionally, we will discuss the integration of real-time data sources from public transportation authorities and other relevant entities, highlighting the technical challenges and solutions encountered during development.

By examining the implementation details, we aim to offer a clear understanding of the practical aspects involved in creating a robust and reliable application that enhances the public transportation experience for users. Through detailed descriptions of the tools and methodologies used, this chapter showcases the technical ingenuity and collaborative effort required to transform our vision into a functional, user-friendly system.

4.2 Development tools

1. **GitHub**:

   GitHub is a web-based platform for version control and collaborative software development. It leverages Git, a distributed version control system, to allow multiple developers to work on projects simultaneously without overwriting each other's contributions.

GitHub provides a centralized location for storing code repositories, tracking changes, managing project documentation, and facilitating collaboration through features like pull requests, issues, and code reviews. It supports open-source projects and private repositories, making it a versatile tool for both individual developers and large teams. Additionally, GitHub integrates with various CI/CD tools and services, enhancing the software development workflow by automating testing, deployment, and other processes[45].

Figure 4.1: GitHub logo

2. **Visual Studio Code**:
   Visual Studio Code (VS Code) is a lightweight, open-source code editor developed by Microsoft. It is designed for a wide range of programming tasks and supports numerous programming languages out of the box. VS Code provides a rich set of features, including syntax highlighting, code completion, debugging, and integrated terminal, which streamline the development process. One of its key strengths is its extensibility through a vast

Figure 4.2: VS Code logo

library of extensions available in the Visual Studio Code Marketplace. These extensions enable developers to customize their coding environment with additional functionality

such as language support, themes, linters, and version control integrations. VS Code also offers powerful collaboration features like Live Share, allowing developers to work together in real-time. Its versatility and robust feature set make it a popular choice among developers for both simple scripting tasks and complex software projects[46].

3. **Google Collaboratory**:

   Google Colab (Colaboratory) is a free, cloud-based platform provided by Google that allows users to write and execute Python code in a Jupyter notebook environment. It is particularly popular for machine learning, data analysis, and educational purposes due to its ease of use and powerful features. One of the primary advantages of Google Colab is its cloud-based environment, eliminating the need for local setup or powerful hardware, as users can leverage Google's cloud infrastructure, including access to GPUs and TPUs for computationally intensive tasks. Additionally, it supports real-time collaboration, enabling multiple users to work on the same notebook simultaneously, which is ideal for team projects and educational settings. The platform integrates seamlessly with Google Drive, allowing users to save and manage their notebooks easily, facilitating smooth data sharing and project management. Furthermore, Google Colab comes with many popular Python libraries pre-installed, reducing setup time and simplifying the development process. Overall, Google Colab is a versatile and accessible tool that supports a wide range of data science and machine learning workflows, making it an invaluable resource for both beginners and experienced practitioners[47].

Figure 4.3: Google Colab logo

4. **Expo**:

Expo is an open-source platform that provides a comprehensive ecosystem of tools for developers building native applications across Android and iOS using JavaScript. It offers a variety of services designed to streamline the development process, including writing, building, updating, submitting, and monitoring mobile apps. Expo simplifies app store submissions, integrates developer tools directly into applications, and accelerates the shipping process. Additionally, it provides a supportive community and access to its development team for assistance. By handling routine and time-consuming tasks, Expo allows developers to concentrate on creating high-quality applications[48].

Figure 4.4: Expo logo

5. **Google Maps Api**:

   The Google Maps API (Application Programming Interface) is a comprehensive suite of web services and tools provided by Google, designed to allow developers to seamlessly integrate Google Maps into their own applications, websites, and services. This API offers a diverse range of functionalities, including the ability to display interactive maps, embed panoramic street views, geocode addresses to obtain geographic coordinates, calculate routes and provide detailed directions, and access real-time traffic data for informed decision-making. It supports extensive customization and manipulation of map data, enabling developers to create highly interactive and dynamic mapping solutions tailored to specific user needs and business requirements. Additionally, the Google Maps API leverages Google's extensive mapping data and robust infrastructure, ensuring reliability, scalability, and the capability to enhance user experiences through location-based services across various platforms. By utilizing the Google Maps API, developers can build powerful, location-aware applications that can significantly improve navigation, spatial analysis, and overall user engagement[49].

Figure 4.5: Google Maps logo

6. **JDK**:

   The Java Development Kit (JDK) is a comprehensive software development package provided by Oracle Corporation for building Java-based applications. It includes a set of development tools, libraries, documentation, and runtime environments necessary for developing, debugging, and deploying Java applications. The JDK encompasses everything a developer needs to write, compile, and run Java code, including the Java Compiler (javac), Java Virtual Machine (JVM), and various utilities and APIs essential for Java programming. This toolkit forms the foundation for Java developers to create robust and platform-independent software solutions across different computing environments[50].

Figure 4.6: JDK logo

7. **JRE**:

   The Java Runtime Environment (JRE) is a software package developed by Oracle Corporation that provides the necessary environment for executing Java applications. It in-

cludes the Java Virtual Machine (JVM), which interprets and executes Java bytecode, as well as core libraries and components required to run Java programs. The JRE is essential for deploying and running Java applications on various platforms and operating systems, ensuring platform independence and compatibility. It does not include development tools like compilers or debuggers, which are provided in the Java Development Kit (JDK), but focuses solely on supporting the runtime execution of Java applications[51].

Figure 4.7: JRE logo

8. **Android SDK**:

   The Android Software Development Kit (SDK) is a collection of tools, libraries, and resources provided by Google for developing applications specifically for the Android platform. It includes a set of development tools such as debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. The Android SDK enables developers to create Android applications using various programming languages such as Java, Kotlin, and C++, and to optimize them for a wide range of Android devices. It provides essential APIs and frameworks that facilitate the development, testing, and deployment of Android apps across different hardware configurations and versions of the Android operating system[52].

Figure 4.8: Android logo

4.3 Development language and framework

4.3.1 Client side

4.3.1.1 React-Native

React Native is an open-source framework developed by Facebook for building native mobile applications using JavaScript and React. It enables developers to create fully functional mobile applications for both Android and iOS platforms from a single codebase. React Native leverages the declarative programming style of React, allowing developers to design complex user interfaces with a component-based architecture.

The framework achieves high performance by rendering components directly to native platform widgets, rather than relying on web-based views. This approach ensures that applications built with React Native have the look and feel of native apps, offering seamless integration with the underlying operating system features. Additionally, React Native supports the use of native code, providing the flexibility to optimize performance-critical sections of the application.

React Native's rich ecosystem of libraries and extensive community support make it a popular choice for mobile app development, enabling rapid iteration and efficient cross-platform development[53].

Figure 4.9: React-Native logo

4.3.1.2 Babel

Babel is a widely-used open-source JavaScript compiler that enables developers to write code using the latest JavaScript features and syntax while ensuring compatibility with older JavaScript

environments. It accomplishes this by transforming ECMAScript 2015+ code into a backwards-compatible version of JavaScript that can be executed in environments that do not natively support newer language features.

Babel supports a wide array of plugins and presets that allow developers to customize their transpilation process, incorporating experimental JavaScript features, JSX syntax used by React, and TypeScript. Additionally, Babel's extensible architecture allows for the development of custom plugins to tailor the transformation process to specific project needs.

By bridging the gap between cutting-edge JavaScript advancements and the diverse landscape of runtime environments, Babel facilitates the use of modern development practices and syntactic enhancements, thereby enhancing developer productivity and code maintainability.

Figure 4.10: Babel logo

4.3.1.3 JavaScript

JavaScript is a high-level, interpreted programming language that is widely used for developing interactive and dynamic web applications. Initially created by Netscape in 1995, JavaScript has evolved into a versatile language that operates on the client-side as well as the server-side, thanks to the development of environments like Node.js.

JavaScript is a core technology of the World Wide Web, alongside HTML and CSS, and is essential for creating responsive user interfaces and enhancing user experiences on websites. It is a prototype-based, multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. JavaScript's dynamic typing, first-class functions, and event-driven architecture enable developers to build complex web applications and perform tasks such as form validation, asynchronous data fetching, and animation.

The language is standardized by the ECMAScript specification (ECMA-262), which ensures consistent implementation across different web browsers. JavaScript's extensive ecosys-

tem includes numerous libraries and frameworks, such as React, Angular, and Vue.js, which streamline the development process and expand its capabilities. As a result, JavaScript remains a fundamental and indispensable tool in modern web development[54]

Figure 4.11: JavaScript logo

4.3.2 Server side

4.3.2.1 Node.js

Node.js is a runtime environment that allows developers to run JavaScript code outside of a web browser. It is built on Chrome's V8 JavaScript engine and operates on a non-blocking, event-driven architecture, which makes it well-suited for scalable and high-performance applications. Node.js enables server-side scripting and facilitates the development of networked applications, web servers, and command-line tools using JavaScript. It provides a rich set of libraries and modules that extend its functionality, making it a versatile platform for building real-time web applications and APIs. Node.js promotes efficient handling of I/O operations and supports asynchronous programming paradigms, enhancing responsiveness and scalability in modern web development[55].

Figure 4.12: Node.js logo

4.3.2.2 Firebase

Firebasase is a comprehensive, cloud-based platform developed by Google that provides a suite of tools and services for building, managing, and scaling web and mobile applications. It offers a range of backend services, including real-time databases, cloud storage, authentication, analytics, and hosting, which enable developers to focus on creating feature-rich applications without the need to manage server infrastructure.

Firebase's core features include the Firebase Realtime Database and Cloud Firestore, which provide scalable and synchronized data storage solutions. Authentication services simplify the process of implementing secure user sign-in methods, supporting various authentication providers such as email, Google, Facebook, and Twitter. Firebase Analytics offers detailed insights into user behavior, helping developers optimize their applications and track key performance indicators.

Other notable services include Firebase Cloud Messaging for push notifications, Firebase Crashlytics for real-time crash reporting, and Firebase Hosting for deploying static websites. Additionally, Firebase integrates seamlessly with other Google Cloud services and supports various platforms, including iOS, Android, and web.

By offering a unified and scalable backend solution, Firebase streamlines the development process, enhances application performance, and provides robust tools for managing and growing applications. This makes it a valuable resource for developers seeking to build high-quality, reliable applications efficiently[56].

Figure 4.13: Firebase logo

4.3.3 Database

4.3.3.1 PostgreSQL

PostgreSQL is an advanced open-source relational database management system (RDBMS) known for its robustness, extensibility, and compliance with SQL standards. It supports a wide range of features, including ACID (Atomicity, Consistency, Isolation, Durability) transactions, complex queries, indexes, and foreign keys, making it suitable for handling complex data relationships and high-volume transactional workloads. PostgreSQL offers extensibility through custom extensions and a rich ecosystem of plugins, allowing developers to add new functionalities seamlessly. It is widely used in enterprise applications, data warehousing, and geospatial applications due to its reliability, performance optimizations, and support for various data types and indexing techniques[57].

Figure 4.14: PostgreSQL logo

4.3.3.2 SQL Language

Structured Query Language (SQL) is a standard programming language designed for managing and manipulating relational databases. It provides a comprehensive set of commands for defining, querying, updating, and administering databases that adhere to the relational model. SQL allows users to create tables, insert and retrieve data, modify data structures, and perform complex queries to extract meaningful information from databases. It is characterized by its declarative nature, where users specify what data, they want to retrieve or manipulate without needing to specify how to achieve it. SQL is essential for interacting with relational database management systems (RDBMS) like MySQL, PostgreSQL, Oracle, and SQL Server, playing a crucial role in data management and application development across various domains including

business, finance, healthcare, and research[58].

Figure 4.15: SQL logo

4.4 Interfaces

In the following, we will present some screenshots of the graphical interfaces developed for our application.

4.4.1 Get started page

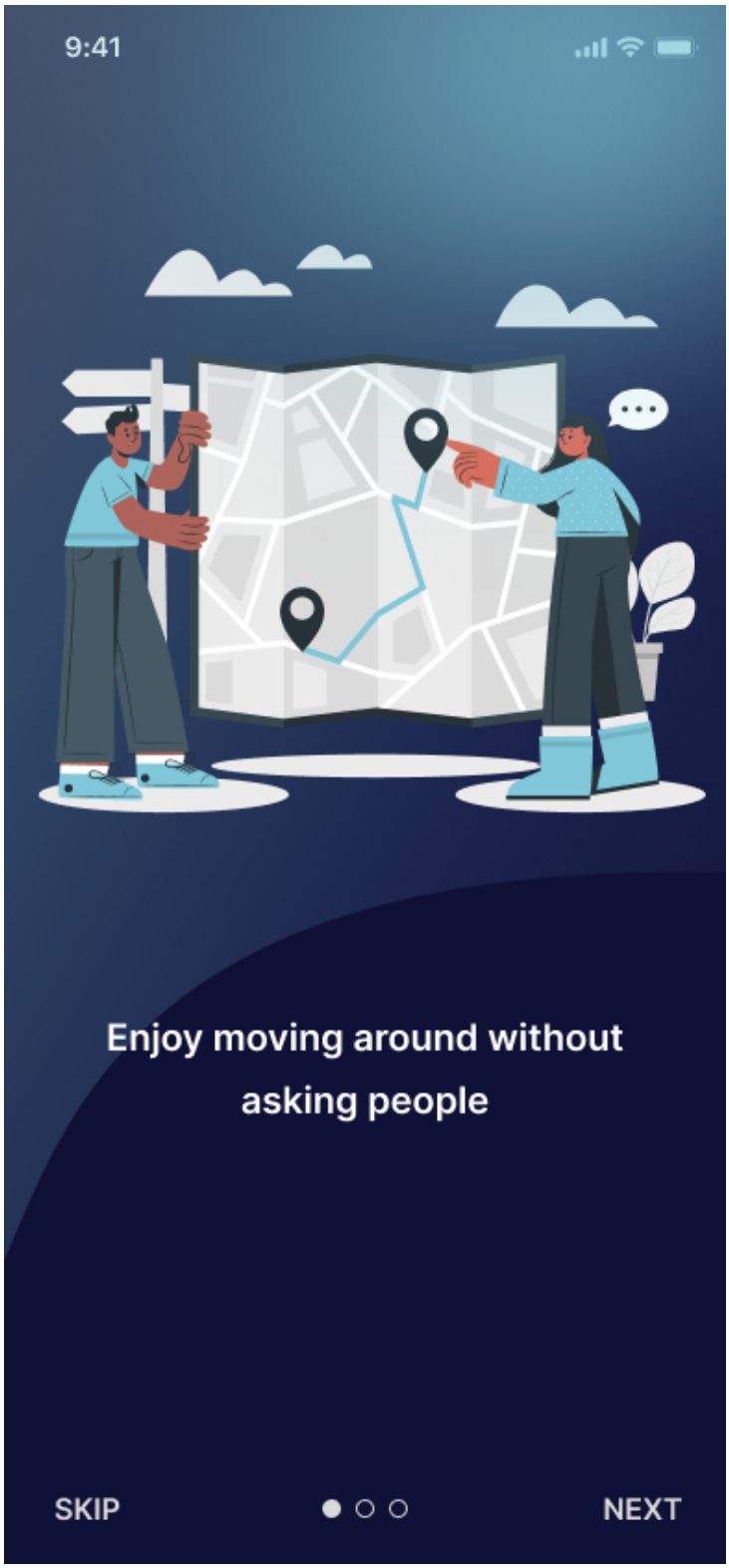


Figure 4.16: Get started 1 interface

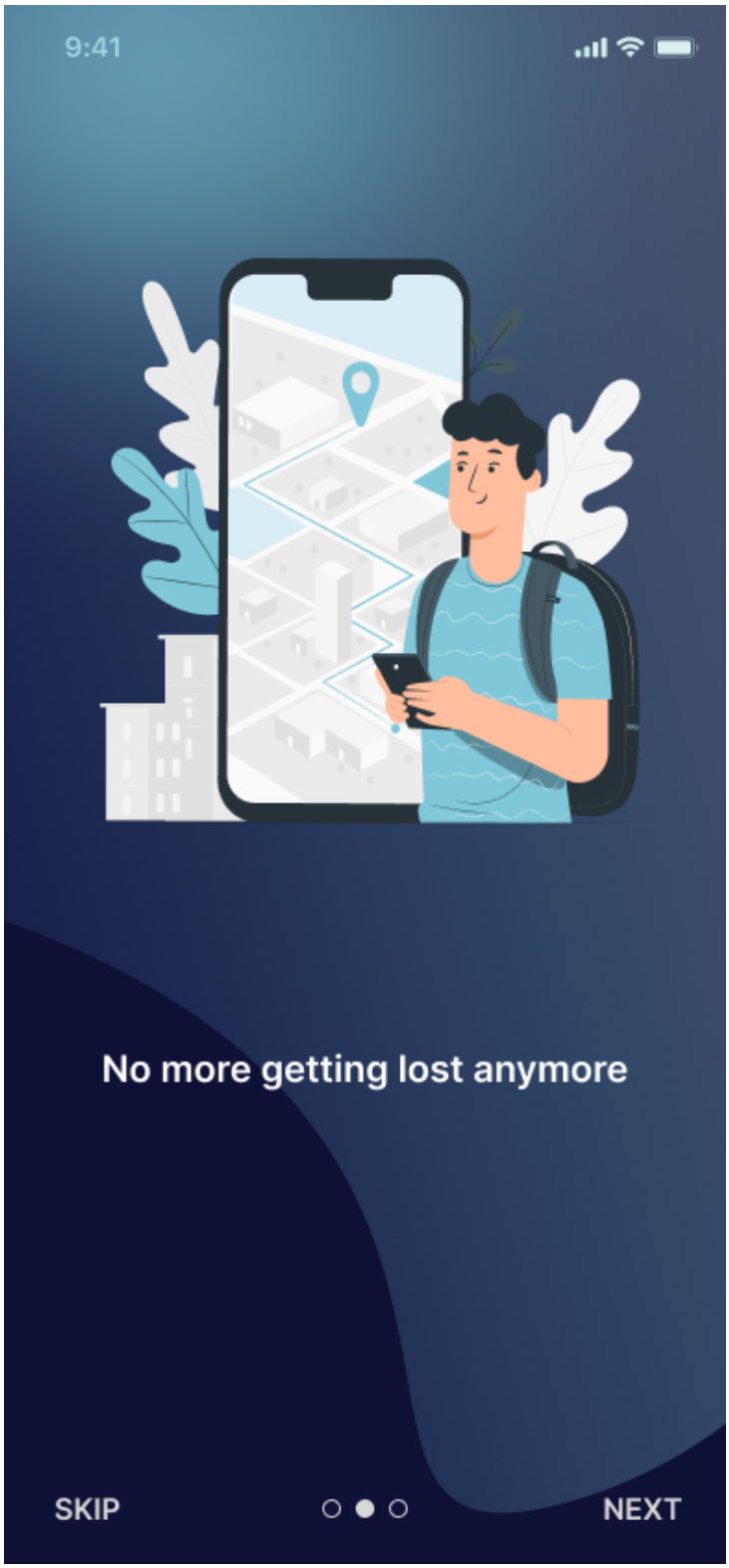


Figure 4.17: Get started 2 interface

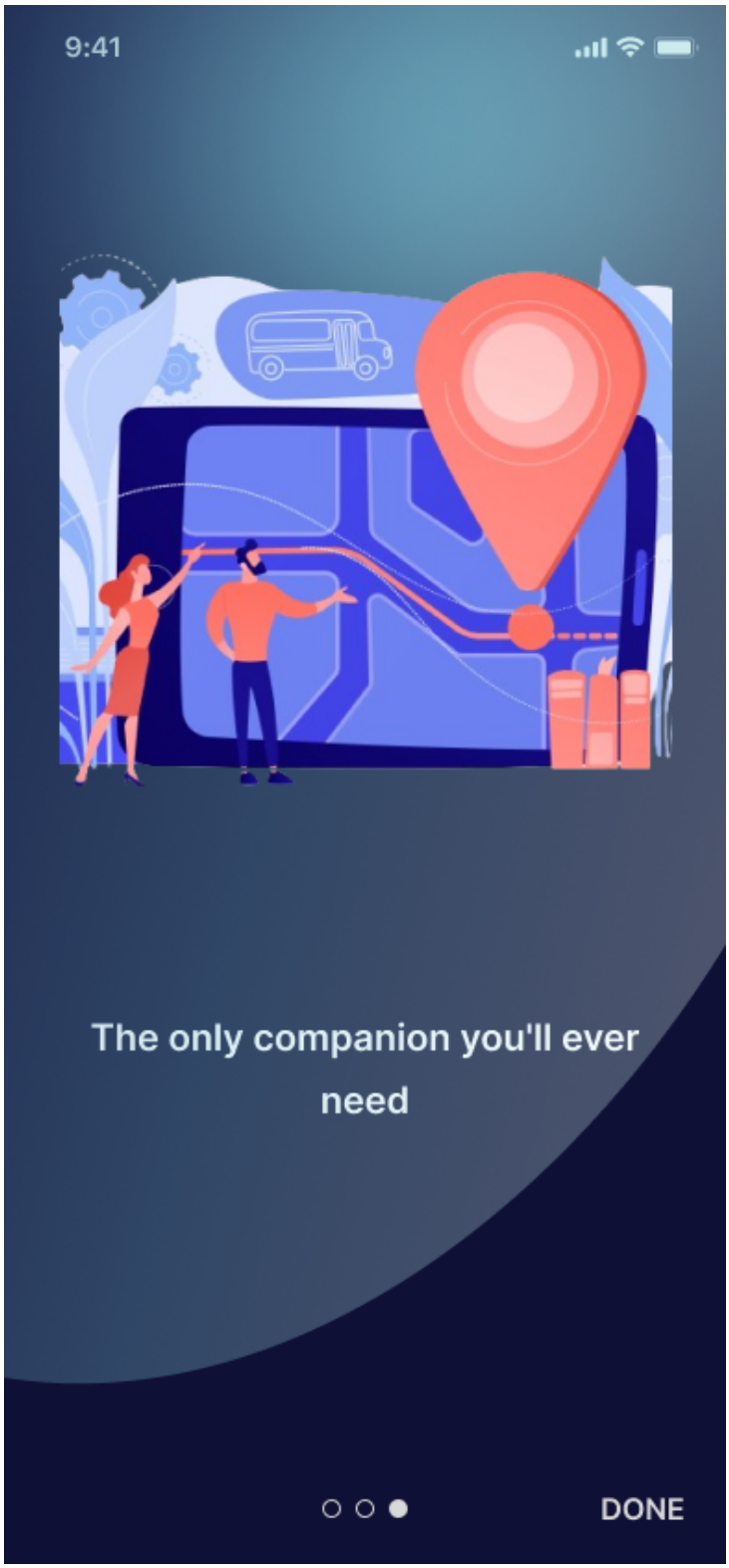


Figure 4.18: Get started 3 interface

These three images appear the first time you open the app, providing a tutorial highlighting the benefits of using our application

4.4.2 Authentication page

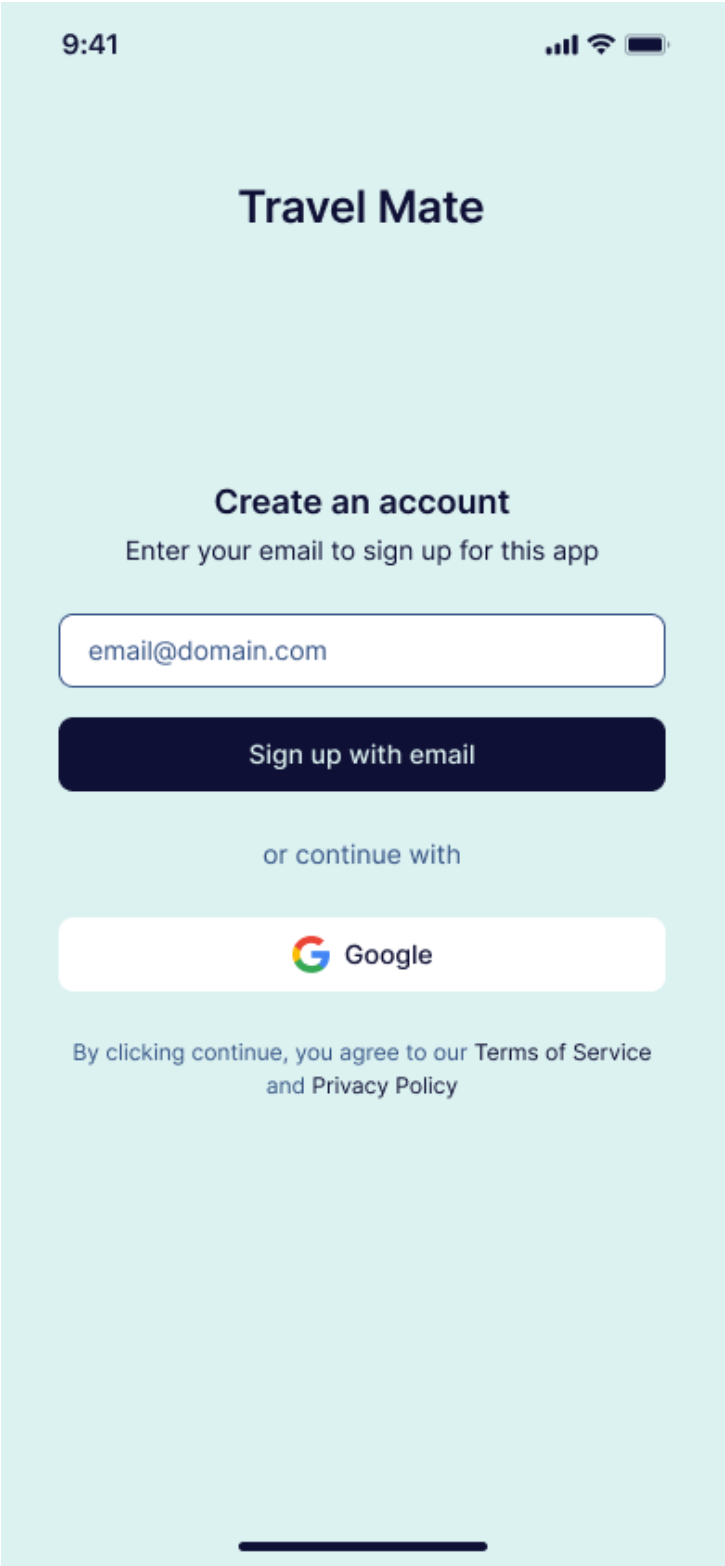


Figure 4.19: Sign up interface

This image depicts the account creation page, presenting various methods to create an account.

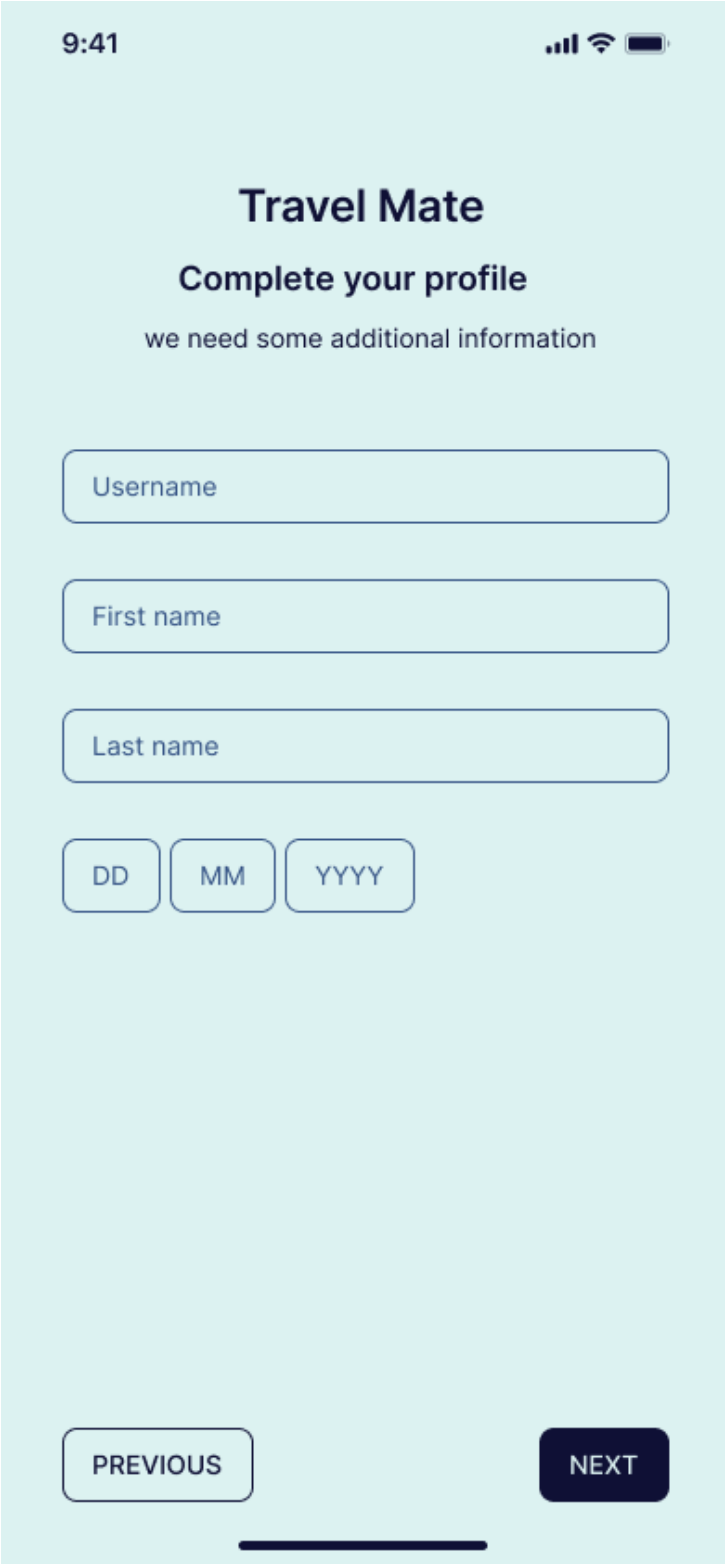


Figure 4.20: Sign up information step interface

This image illustrates the personal information page to finalize your account creation.

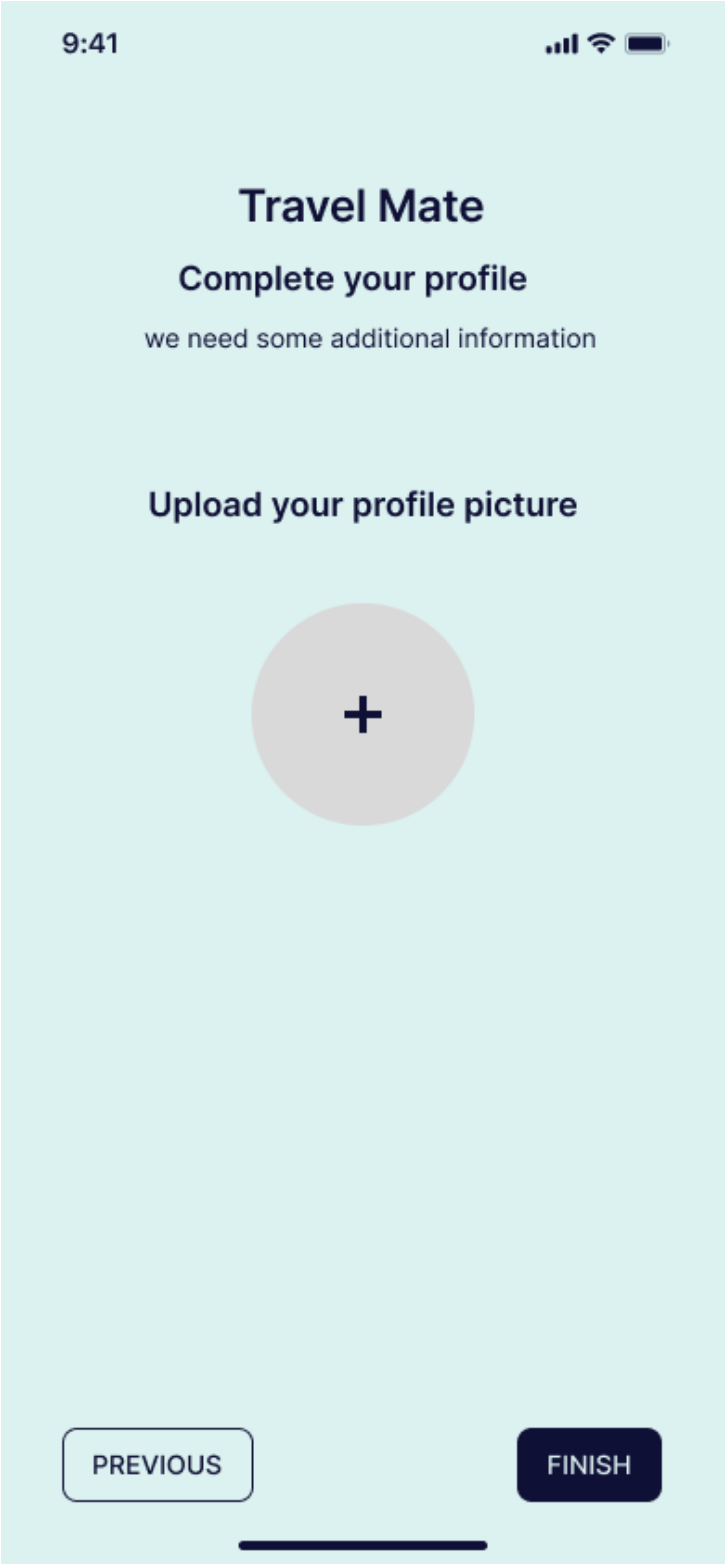


Figure 4.21: Sign up picture step interface

This image shows a profile creation section where you can upload your profile picture

4.4.3 Home page

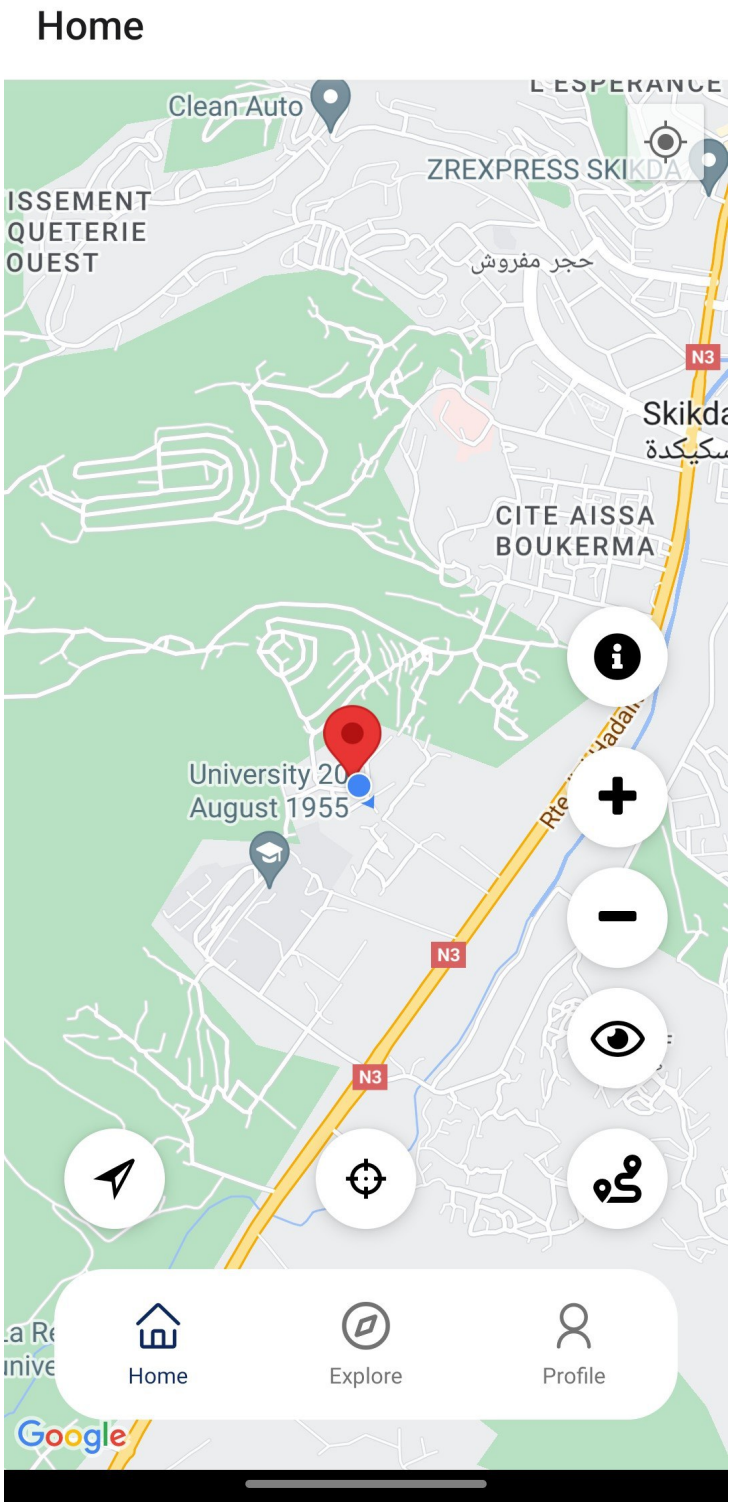


Figure 4.22: Home page

This figure depicts the home page of our application. From here, you can navigate the map to set your current location and destination. The home page features a guide on using the applica-

tion, additional information, zoom controls, options to view various transportation methods in your area, a routing button, destination selection, and a navigation bar for easy access to home, profile, or explore sections.

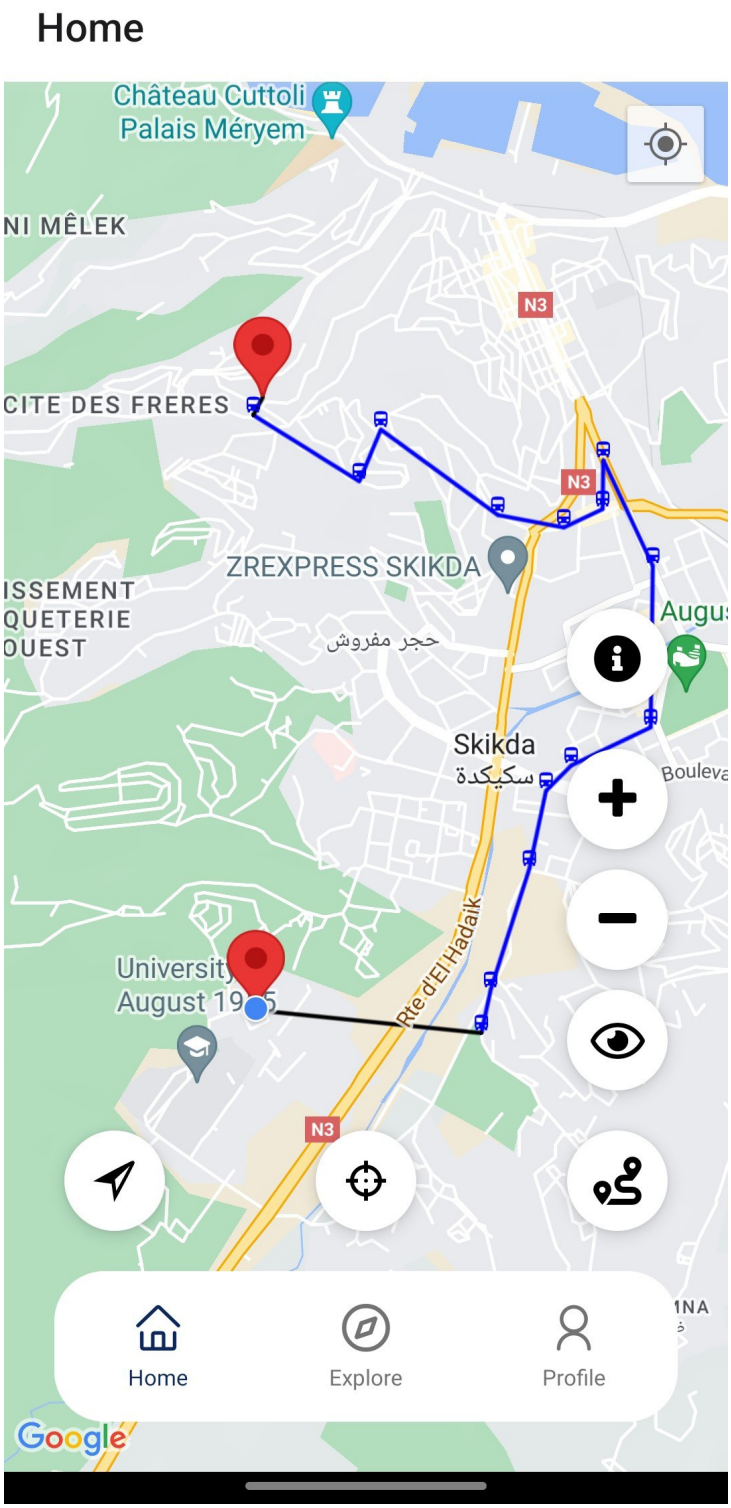


Figure 4.23: Finding the short path interface

This figure illustrates the shortest bus route after selecting the starting and destination loca-

tions, also displaying bus stops along the way

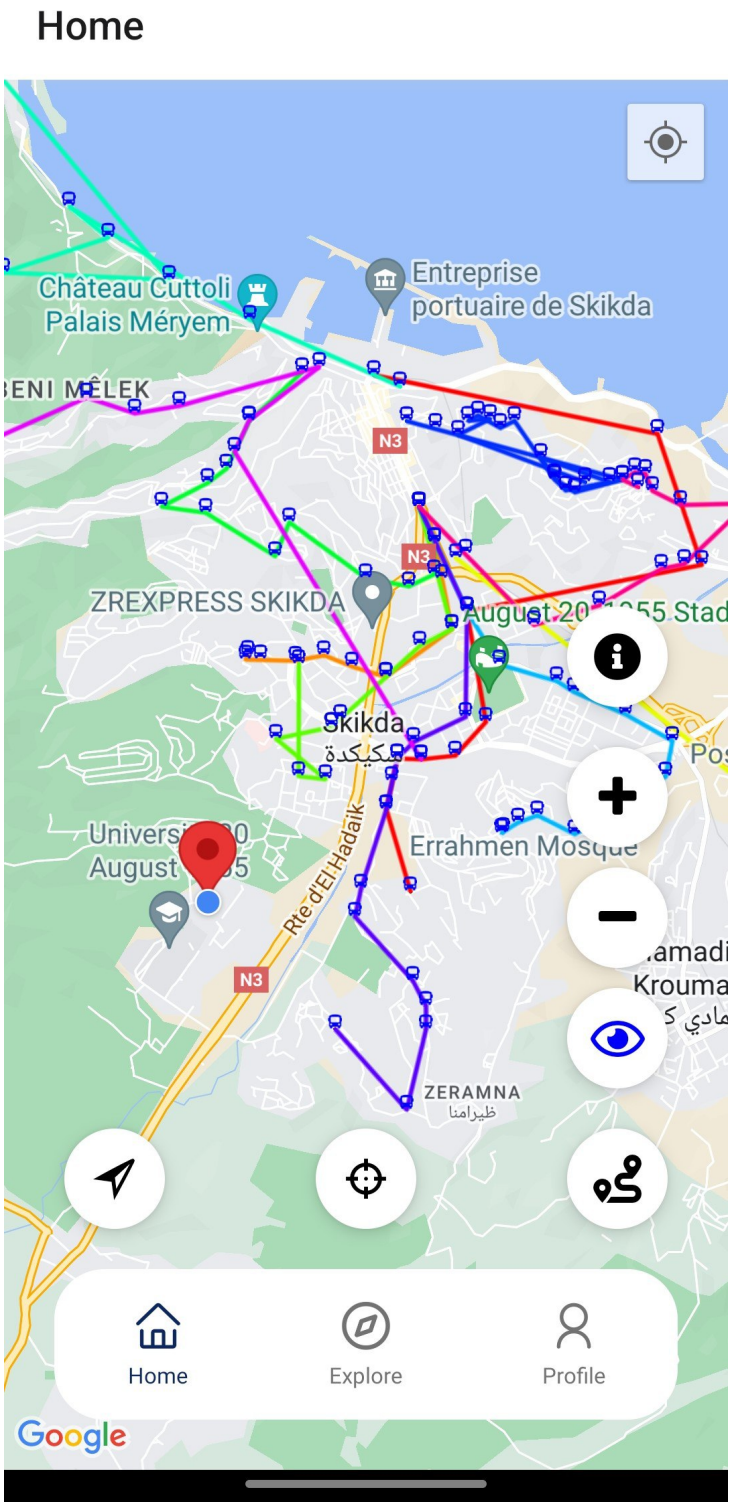


Figure 4.24: All available transportation means interface

This figure displays all available transportation means and their respective routes

4.5 Conclusion

In conclusion, this chapter has provided an in-depth look into the implementation phase of our public transportation optimization application. We have detailed the various tools and technologies utilized to transform our conceptual framework into a functional and user-friendly system, illustrating the technical ingenuity and collaborative effort involved in the development process.

The integration of real-time data sources from public transportation authorities and other relevant entities was also explored, highlighting the technical challenges and solutions encountered. The use of development tools such as GitHub, Visual Studio Code, Google Colaboratory, Expo, Google Maps API, JDK, JRE, and Android SDK played a crucial role in building the application's robust infrastructure.

On the client side, React Native and supporting technologies like Babel and JavaScript were employed to create a seamless user experience, while server-side components like Node.js and Firebase provided the necessary backend support. The combination of these technologies ensured that the application is efficient, scalable, and capable of providing real-time updates and personalized guidance to users.

Furthermore, screenshots of the application have been included to provide a visual representation of the user interface and functionality. The detailed descriptions of the tools and methodologies used, along with these screenshots, offer a clear understanding of the practical aspects involved in creating this public transportation optimization application. The result is a reliable, efficient, and user-centric system that enhances the public transportation experience by reducing wait times, optimizing travel routes, and increasing accessibility for all users.

General conclusion

This project has navigated the dynamic landscape of public transportation optimization by introducing an innovative application tailored for citizens. This research has centered on enhancing urban services through advanced technological strategies, aiming to seamlessly integrate into users' daily mobility routines. Throughout the exploration across chapters, we have delved deeply into core optimization principles in public transportation, the intricate technical framework of the application, and the nuanced challenges and opportunities inherent in its development.

This project highlights the pivotal role of innovation in shaping the future of public transportation, where personalization emerges as a key component of user satisfaction. The application we have developed holds promise in meeting the diverse needs of travelers and residents alike, offering an efficient solution to enhance public transportation services.

Looking forward, this project signifies a progressive approach to urban mobility, harnessing the power of artificial intelligence to refine and optimize public transportation operations. By prioritizing user-centric solutions, this initiative aims to evolve alongside the dynamic demands of modern commuters, thereby contributing to a more accessible, efficient, and sustainable urban transportation ecosystem.

Bibliography

[1] S. Guze, "Graph theory approach to transportation systems design and optimization," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 8, 2014.

[2] S. Pirzada, "Applications of graph theory," in *PAMM: Proceedings in Applied Mathematics and Mechanics*, (Berlin), pp. 2070013–2070013, WILEY□VCH Verlag, 2007.

[3] R. E. Korf and P. Schultze, "Large-scale parallel breadth-first search," in *AAAI*, pp. 1380–1385, 2005.

[4] A. V. Goldberg and C. Harrelson, "Computing the shortest path: A search meets graph theory," in *SODA*, pp. 156–165, 2005.

[5] Y.-J. Chang, V. Dani, T. P. Hayes, *et al.*, "The energy complexity of bfs in radio networks," in *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pp. 273–282, 2020.

[6] V. Vardhini, "Breadth first search," 2021. Figure.

[7] M. Sniedovich, "Dijkstra's algorithm revisited: the dynamic programming connexion," *Control and Cybernetics*, vol. 35, no. 3, pp. 599–620, 2006.

[8] D. R. Lanning, G. K. Harrell, and J. Wang, "Dijkstra's algorithm and google maps," in *Proceedings of the 2014 ACM Southeast Regional Conference*, pp. 1–3, 2014.

[9] Z. Zhu, L. Li, W. Wu, *et al.*, "Application of improved dijkstra algorithm in intelligent ship path planning," in *2021 33rd Chinese Control and Decision Conference (CCDC)*, pp. 4926–4931, IEEE, 2021.

[10] X. Zou, "Algorithm shortest path," 2018. Figure.

[11] R. D. Gunawan, R. Napianto, R. I. Borman, *et al.*, "Implementation of dijkstra's algorithm in determining the shortest path (case study: Specialist doctor search in bandar lampung)," *Int. J. Inf. Syst. Comput. Sci*, vol. 3, no. 3, pp. 98–106, 2019.

[12] V. Gogoncea, G. Murariu, and L. Georgescu, "The use of dijkstra's algorithm in waste management problem," *Annals of" Dunarea de Jos" University of Galati, Fascicle V, Technologies in machine building*, vol. 28, no. 2, pp. 125–127, 2010.

[13] "A* algorithm example: Is it correct?," 2012. Figure.

[14] G. Barbarosoglu and D. Ozgur, "A tabu search algorithm for the vehicle routing problem," *Computers Operations Research*, vol. 26, no. 3, pp. 255–270, 1999.

[15] J. P. Marques-Silva and K. A. Sakallah, "Grasp: A search algorithm for propositional satisfiability," *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 506–521, 1999.

[16] X.-S. Yang, "Metaheuristic optimization," *Scholarpedia*, vol. 6, no. 8, p. 11472, 2011.

[17] J. A. Parejo, A. Ruiz-Cortés, S. Lozano, *et al.*, "Metaheuristic optimization frameworks: a survey and benchmarking," *Soft Computing*, vol. 16, pp. 527–561, 2012.

[18] A. G. Nikolaev and S. H. Jacobson, "Simulated annealing," in *Handbook of metaheuristics*, pp. 1–39, 2010.

[19] "Simulated annealing optimization of a one-dimensional objective function," 2016. Figure.

[20] F. Y. Vincent, S.-W. Lin, W. Lee, *et al.*, "A simulated annealing heuristic for the capacitated location routing problem," *Computers Industrial Engineering*, vol. 58, no. 2, pp. 288–299, 2010.

[21] A. Stevanovic, P. T. Martin, and J. Stevanovic, "Vissim-based genetic algorithm optimization of signal timings," *Transportation Research Record*, vol. 2035, no. 1, pp. 59–68, 2007.

[22] S. N. Sivanandam, S. N. Deepa, S. N. Sivanandam, *et al.*, *Genetic algorithms*. Springer Berlin Heidelberg, 2008.

[23] "Social learning algorithms: Particle swarm optimization (pso)," 2013. Figure.

[24] A. G. Gad, “Particle swarm optimization algorithm and its applications: a systematic review,” *Archives of computational methods in engineering*, vol. 29, no. 5, pp. 2531–2561, 2022.

[25] K. E. Parsopoulos and M. N. Vrahatis, “Particle swarm optimization method in multiobjective problems,” in *Proceedings of the 2002 ACM symposium on Applied computing*, pp. 603–607, 2002.

[26] “The operation of the ant colony algorithm,” 2019. Figure.

[27] C. Blum, “Ant colony optimization: Introduction and recent trends,” *Physics of Life Reviews*, vol. 2, no. 4, pp. 353–373, 2005.

[28] X. jun Liu, H. Yi, and Z. hua Ni, “Application of ant colony optimization algorithm in process planning optimization,” *Journal of Intelligent Manufacturing*, vol. 24, pp. 1–13, 2013.

[29] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[30] Y. Li, “Deep reinforcement learning: An overview,” *arXiv preprint arXiv:1701.07274*, 2017.

[31] K. Arulkumaran, M. P. Deisenroth, M. Brundage, *et al.*, “Deep reinforcement learning: A brief survey,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[32] T. D. Science, “Reinforcement learning 101,” 2018. Figure.

[33] V. François-Lavet, P. Henderson, R. Islam, *et al.*, “An introduction to deep reinforcement learning,” *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[34] DataCamp, “A comprehensive introduction to graph neural networks (gnns) tutorial,” 2020. Figure.

[35] W. Jiang and J. Luo, “Graph neural network for traffic forecasting: A survey,” *Expert Systems with Applications*, vol. 207, p. 117921, 2022.

[36] R. Zhu, K. Zhao, H. Yang, *et al.*, "Aligraph: A comprehensive graph neural network platform," *arXiv preprint arXiv:1902.08730*, 2019.

[37] S. Porru, F. E. Misso, F. E. Pani, *et al.*, "Smart mobility and public transport: Opportunities and challenges in rural and urban areas," *Journal of Traffic and Transportation Engineering (English edition)*, vol. 7, no. 1, pp. 88–97, 2020.

[38] P. Davidsson, B. Hajinasab, J. Holmgren, *et al.*, "The fourth wave of digitalization and public transport: Opportunities and challenges," *Sustainability*, vol. 8, no. 12, p. 1248, 2016.

[39] J. Bharadiya, "Artificial intelligence in transportation systems a critical review," *American Journal of Computing and Engineering*, vol. 6, no. 1, pp. 34–45, 2023.

[40] R. Abduljabbar, H. Dia, S. Liyanage, *et al.*, "Applications of artificial intelligence in transport: An overview," *Sustainability*, vol. 11, no. 1, p. 189, 2019.

[41] N. P. Farazi, B. Zou, T. Ahamed, *et al.*, "Deep reinforcement learning in transportation research: A review," *Transportation Research Interdisciplinary Perspectives*, vol. 11, p. 100425, 2021.

[42] P. Almasan, J. Suárez-Varela, K. Rusek, *et al.*, "Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case," *Computer Communications*, vol. 196, pp. 184–194, 2022.

[43] X. Hu, C. Zhao, and G. Wang, "A traffic light dynamic control algorithm with deep reinforcement learning based on gnn prediction," *arXiv preprint arXiv:2009.14627*, 2020.

[44] S. Munikoti, D. Agarwal, L. Das, *et al.*, "Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[45] K. Peterson, "The github open-source development process," tech. rep., http://kevinp.me/github-process-research/github-processresearch.pdf, 2013. visited on 05/11/2017.

[46] V. S. Code, "Visual studio code," tech. rep., https://reactnative.dev/, 2019. visited on November 2, 2019.

[47] E. Bisong and E. Bisong, "Google colaboratory," in *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, pp. 59–64, 2019.

[48] H. Hutri, "Comparison of react native and expo," tech. rep., 2023.

[49] S. Kobayashi, T. Fujioka, Y. Tanaka, *et al.*, "A geographical information system using the google map api for guidance to referral hospitals," *Journal of Medical Systems*, vol. 34, pp. 1157–1160, 2010.

[50] J. Gößner, P. Mayer, and F. Steimann, "Interface utilization in the java development kit," in *Proceedings of the 2004 ACM symposium on Applied computing*, pp. 1310–1315, 2004.

[51] J. F. Schommer, D. Franke, S. Kowalewski, *et al.*, "Evaluation of the real-time java runtime environment for deployment in time-critical systems," in *Proceedings of the 7th International Workshop on Java Technologies for Real-Time and Embedded Systems*, pp. 51–60, 2009.

[52] M. Linares-Vásquez, G. Bavota, M. D. Penta, *et al.*, "How do api changes trigger stack overflow discussions? a study on the android sdk," in *Proceedings of the 22nd International Conference on Program Comprehension*, pp. 83–94, 2014.

[53] R. Native, "React native," tech. rep., https://reactnative.dev/, 2020. visited on November 2, 2019.

[54] A. Guha, C. Saftoiu, and S. Krishnamurthi, "The essence of javascript," in *ECOOP 2010–Object-Oriented Programming: 24th European Conference, Maribor, Slovenia, June 21-25, 2010. Proceedings*, pp. 126–150, 2010.

[55] M. Cantelon, M. Harter, T. J. Holowaychuk, *et al.*, *Node. js in Action*. Greenwich: Manning, 2014.

[56] C. Khawas and P. Shah, "Application of firebase in android app development-a study," *International Journal of Computer Applications*, vol. 179, no. 46, pp. 49–53, 2018.

[57] G. Smith, *PostgreSQL 9.0: High Performance*. Packt Publishing Ltd, 2010.

[58] J. Melton, "Sql language summary," *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 141–146, 1996.

جامعة 20 أوت سكيكدة
كلية العلوم
قسم الإعلام الآلي

مشروع لنيل شهادة ماستر + شهادة مؤسسة ناشئة في اطار القرار الوزاري 1275

تطبيق التنقل الذكي: تحسين عمليات التنقل بوسائل النقل العام باستخدام الذكاء الاصطناعي

من إعداد الطلبة:

بوربيع محمد نبيل
بوفارس صبري
ناجي سعيد فاضل حسن

لجنة المناقشة:

رئيس اللجنة: الأستاذ المشرف: د.حزمون سميرة
الأستاذ مساعد المشرف: د.بوغموزة فاتح
الأستاذ الممتحن: د.بوالريش حنان
خبير BMC:
الشريك الإقتصادي:

السنة الدراسية: 2023/2024

الفهرس

قائمة الصور

قائمة الجداول

إهداء

نهدي ثمرة جهدنا المتواضعة الى الذين كانوا سببا في نجاحنا

والدينا اطال الله في عمرهما

إلى إخوتنا و أخواتنا كل بإسمه

إلى جميع أهلنا و أقاربنا

إلى كل من علمنا حرفا طيلة مسارنا الدراسي من التعليم الإبتدائي الى الجامعي أساتذتنا الكرام

إلى زملاءنا و زميلاتناالذين جمعتنا بهم صحبة الدراسة

إلى كل من ساعدنا بالقول والفعل وكان سندا لنا في إنجاز هذا العمل

إلى كل من يجاهد ويناضل من أجل تحصيل العلم

إلى كل من يكن لنا الحب والإحترام والتقدير.

شكر و عرفان

قال تعالى: رَبِّ أَوْزِعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ الَّتِي أَنْعَمْتَ عَلَيَّ وَعَلَىٰ وَالِدَيَّ وَأَنْ أَعْمَلَ صَالِحًا تَرْضَاهُ وَأَدْخِلْنِي بِرَحْمَتِكَ فِي عِبَادِكَ الصَّالِحِينَ (19) سورة النمل

جزيل الشكر والحمد لله عز وجل الذي أنعم علينا بإتمام هذا العمل المتواضع، له الحمد كما ينبغي لجلال وجهه ولعظيم سلطانه.

نتقدم بالشكر الجزيل إلى الأستاذة المشرفة **"حزمون سميرة"** و الأستاذ المشرف **"بوغموزة فاتح"** اللذان رافقانا طيلة هذا المشروع وأثمرانا بالمعلومات والنصائح القيمة، راجيين من الله عز وجل أن يسدد خطاياهما ويحقق أمانيهما فجزاهما الله عنا كل خير.

وأيضًا نتقدم بالشكر الجزيل لكل مسؤولي وعمال كل من الحاضنة ودار المقاولاتية لجامعة 20 أوت 1955 سكيكدة على المساعدات والتسهيلات المقدمة، فجزاهم الله عنا كل خير.

إلى جميع أساتذة قسم الإعلام الآلي كل بمقامه، وإلى كل من ساعدنا من قريب أو من بعيد في إنجاز هذا العمل المتواضع.

وفي الختام نتوجه بالشكر لأساتذة أعضاء لجنة المناقشة الأفاضل لقبولهم مناقشة هذه المذكرة.

مقدمة

في عصرنا الحالي الذي يتسم بسرعة الحياة وارتفاع عدد السكان في المدن، أصبحت وسائل النقل العامة أمرًا لا غنى عنه للتنقل بين الأماكن بسرعة وكفاءة. تعاني المدن الكبرى من ازدحام مروري خانق، مما يجعل الاعتماد على السيارات الخاصة أمراً غير مجدٍ في الكثير من الأحيان، سواء من حيث الوقت أو التكلفة. بالإضافة إلى ذلك، يسهم الاعتماد على وسائل النقل العامة في تقليل الانبعاثات الكربونية والحفاظ على البيئة، مما يجعلها خيارًا مستدامًا للمستقبل.

ومع ذلك، يواجه العديد من الأشخاص صعوبة في فهم الخدمات المتاحة وكيفية الوصول إليها، مما يجعل التنقل يتطلب تخطيطًا مسبقًا واستفسارًا مستمرًا من الآخرين. يتضاعف هذا التحدي بالنسبة للزوار الجدد للمدينة أو السياح الذين قد لا يكونون على دراية بالنظام المحلي للنقل، مما يزيد من الحاجة إلى حل يسهل عليهم الوصول إلى المعلومات الضرورية.

هنا يأتي دور تطبيق **'Travel Mate'**، الذي يهدف إلى تبسيط وتحسين تجربة السفر باستخدام وسائل النقل العامة. يتيح نظامنا للمستخدمين الوصول السريع إلى معلومات حية حول جداول الحافلات، ومواقف القطارات، والمترو، وغيرها من وسائل النقل، دون الحاجة إلى البحث أو الاستفسار من الآخرين. لا يقتصر دور التطبيق على توفير الجداول الزمنية فحسب، بل يمتد ليشمل خدمات إضافية مثل التنبيهات المبكرة عن التأخيرات والإغلاقات، وإرشادات الطريق الدقيقة التي تساعد المستخدمين في تخطيط رحلاتهم بفعالية.

باستخدام تقنيات حديثة مثل تحليل البيانات والذكاء الاصطناعي، يمكن لتطبيق **'Travel Mate'**، توفير تجربة سفر سلسة وموثوقة للمستخدمين، مما يسهل عليهم الوصول إلى وجهاتهم بكل يسر وراحة. كما يتعلم التطبيق من نمط تنقلات المستخدمين ويوفر لهم توصيات مخصصة، مما يزيد من كفاءة التنقل ويقلل من زمن الانتظار. إن مهمتنا هي تحويل التنقل اليومي إلى تجربة ممتعة وميسرة، دون الحاجة إلى التوجيهات اليدوية أو البحث عن المعلومات، مع تقديم دعم مستمر للمستخدمين من خلال تحديثات فورية وإشعارات متزامنة مع الأحداث الجارية.

بفضل **'Travel Mate'**، لم يعد التنقل في المدينة تحديًا مرهقًا بل أصبح عملية سلسة ومدروسة تضمن الوصول إلى الوجهة بأقل جهد ممكن. نحن نؤمن بأن توفير وسائل نقل عامة فعالة ومعلومات دقيقة يمكن أن يحدث فارقًا كبيرًا في حياة الأفراد، مما يعزز من جودة الحياة ويسهم في بناء مجتمعات أكثر تواصلاً واستدامة.

تناول المحور الأول تقديم المشروع الذي يهدف إلى تطوير تطبيق للنقل العام يسهل التنقل ويشمل التخطيط والتحضير والتصميم والتطوير والدعم المستمر. يتناول المحور الثاني جوانب الابتكار بما في ذلك القيم المقترحة و المضافة التي تميزنا عن غيرنا. المحور الثالث يتناول التحليل الاستراتيجي للسوق بما في ذلك عرض القطاع السوقي، قياس شدة المنافسة، والاستراتيجية التسويقية. المحور الرابع يتناول خطة الإنتاج والتنظيم بدءًا من التخطيط والتصميم حتى الدعم والتحسين المستمر. المحور الخامس يتناول الخطة المالية للمشروع والذي تناونا فيه خطة المالية للمشروع ونموذج العمل التجاري وفية التكاليف والرسوم النظرة التفاؤلية والتشاؤمية وخطة الخزينه، بينما المحور السادس يركز على النموذج التجريبي الأولي، والمحور السابع يتناول نموذج العمل التجاري.

المحور 1 تقديم المشروع

1.1 تمهيد

إن تقديم فكرة مشروع مبتكر تتطلب عناية وتفكير دقيق، إذ يجب أن تتمتع الفكرة بصفات محددة، تسهم في تحقيق النجاح كما ينبغي أن تكون فكرة مبتكرة وتقدم حلاً فريًدا لمشاكل متراكمة وتلبي حاجة واضحة في المجتمع. ويتعين أن تتميز الفكرة عن المشاريع الموجودة بالفعل من خلال عناصر جديدة ومنهجية فريدة تضيف قيمة حقيقية للمستهلكين، بالإضافة إلى ذلك، يجب أن تكون الفكرة قابلة للتنفيذ والاستدامة مع توفر الموارد المالية والبشرية والتكنولوجية اللازمة لتحقيقها بفعالية. أما من الناحية الربحية فيجب أن تكون الفكرة قادرة على تحقيق عائد مالي ملموس حيث تم التركيز في هذا المحور على طرح فكرة مشروعنا، وفريق العمل الذي سيتولى تنفيذه، وتحديد الأهداف المرجوة من المشروع، بالإضافة الى وضع جدول زمني لتحقيقها.

2.1 فكرة المشروع

يتضمن المشروع تطوير نظام متطور يمكّن المستخدمين من الوصول إلى خدمات النقل العام والاستفادة منها دون الحاجة إلى استفسارات أو مساعدة يدوية. يستفيد هذا النظام من التقنيات المتقدمة، مثل تحليل البيانات في الوقت الفعلي وخوارزميات التعلم الآلي، لتزويد المستخدمين بمعلومات دقيقة وحديثة حول خيارات النقل العام من مسارات و نقاط توقف والجداول الزمنية.

1.2.1 كيف بدأت الفكرة

برزت الحاجة إلى هذا المشروع نتيجة توسع المدن وازدياد عدد السكان بشكل هائل، مما أدى إلى ضغط هائل على أنظمة النقل داخل المدن. وأصبح من الضروري توفير أدوات متطورة تتناسب مع هذا النمو وتلبي احتياجات المسافرين، الذين باتوا بحاجة ماسة إلى معلومات دقيقة وجداول محدثة لوسائل النقل العام. ويهدف هذا المشروع إلى تلبية هذه الاحتياجات من خلال تقديم توجيهات دقيقة وفعالة لتحسين تجربة التنقل للمسافرين.

يهدف هذا التطبيق إلى حل هذه المشكلة من خلال توفير تطبيق شامل يوفر للمستخدمين تخطيط رحلات مخصصة بناءً على احتياجات المستخدم وتفضيلاته. سيتم دمج التطبيق مع أنظمة النقل المختلفة لجمع البيانات في الوقت الفعلي حول جداول الحافلات ومواقف القطارات وأوقات الوصول والمغادرة. بالإضافة إلى ذلك، سيوفر التطبيق ميزات إضافية مثل:

- تنبيهات مبكرة في حال حدوث تأخيرات أو إلغاءات
- معلومات حية ودقيقة حول جميع وسائل النقل العام في منطقتهم
- إرشادات طريق دقيقة باستخدام خرائط GPS وبيانات المرور الحية
- تقديم حلول تنقل مُدمجة تُراعي رغبات كل شخص، من خلال دمج خيارات النقل المتوفرة في المنطقة كالمشي، حافلات، سيارات أجرة، ميترو...، مع التركيز على تقليل التكلفة أو تقصير مدة الرحلة.
- تقديم ملاحظات واقتراحات لتحسين الخدمة

2.2.1 ما الذي سوف نقوم به

1. **تصميم وتطوير تطبيق الهاتف المحمول**:
   - تصميم واجهة مستخدم سهلة الاستخدام وبديهية.
   - دمج خرائط GPS.
   - تطوير خوارزميات ذكية لمعالجة البيانات وتحليلها.
   - ربط التطبيق بأنظمة النقل المختلفة لجمع البيانات في الوقت الفعلي.
2. **جمع البيانات ودمجها**:
   - التعاون مع شركات ومؤسسات النقل العام ومقدمي الخدمات الآخرين لجمع البيانات.
   - ضمان دقة البيانات وتحديثها بشكل مستمر.
   - دمج البيانات من مصادر مختلفة (مثل أنظمة النقل، وبيانات المرور، والطقس).
3. **تطوير خوارزميات الذكاء الاصطناعي**:
   - استخدام تقنيات التعلم الآلي لتوقع التأخيرات والازدحام المروري.
   - تقديم اقتراحات مسار الرحلة المُحسّنة بناءً على احتياجات المستخدم.
   - تحسين دقة المعلومات وتقديمها بطريقة مفهومة.
   - استخدام خوارزميات التعلم المعزز لفهم سلوك كل مستخدم وتقديم تجربة مُخصصة له.
   - تطوير خوارزميات التعلم المعزز من أجل الاستفادة من آراء المستخدمين وتقييماتهم للرحلات المقترحة، وذلك بهدف تحسين جودة الخدمات المقدمة وتعزيز تجربة العملاء.
4. **اختبار التطبيق وإطلاقه**:

- اختبار التطبيق بشكل شامل لضمان دقة المعلومات وسهولة الاستخدام.
- الحصول على ملاحظات من المستخدمين وإجراء التعديلات اللازمة.
- إطلاق التطبيق على متاجر التطبيقات وجعله متاحًا للجمهور

3.2.1 القيم المقترحة

المشروع يهدف إلى إنشاء تطبيق يسهل على المستخدمين استغلال وسائل النقل العامة دون الحاجة إلى استشارة الآخرين. يتمثل الهدف في إمكانية استغلال وسائل النقل العمومي بأنجع الطرق وتيسير عملية التنقل، وذلك من خلال استخدام تكنولوجيات الذكاء الاصطناعي لتوفير معلومات دقيقة وموثوقة وإرشادات مخصصة لكل مسافر. حيث تتميز القيم المقترحة بالآتي:

- سهولة الاستخدام: توفير واجهة مستخدم بسيطة وواضحة تجعل من السهل على المستخدمين الوصول إلى المعلومات واستخدام النظام.
- توفير اقتراحات مخصصة للمستخدمين بناءا على محيطهم وتجاربهم السابقة: يعتمد التطبيق الذكي على تحليل سلوك المستخدم وتفضيلاته، مثل الاعتماد على سيارات الأجرة، تفضيل المشي، أو اختيار وسائل النقل الاقتصادية، لتقديم اقتراحات مخصصة توفر أفضل الطرق والخدمات الملائمة لتفضيلاتهم واحتياجاتهم الفردية.
- الدقة والموثوقية: تعتمد في التطبيق على تقديم معلومات دقيقة حول أوقات الوصول والانطلاق والمسارات التوجيهية والتكلفة الشاملة للرحلة من ناحية الوقت او الثمن، لضمان تجربة مستخدم مثالية وموثوقة.
- الكفاءة: تقديم أفضل الطرق والتوجيهات لتحقيق رحلات أكثر كفاءة واقتصادًا.
- التفاعلية: إمكانية التفاعل مع التطبيق بسهولة.
- التحديثات الدورية: تحديثات منتظمة للتطبيق لتحسين الأداء وتوفير أحدث المعلومات للمستخدمين.
- الهدف النهائي: تحقيق تجربة تنقل مريحة وفعالة للمسافرين.

3.1 فريق العمل

يتكون فريق المشروع من الطلبة بوربيع محمد نبيل، سعيد فاضل حسن ناجي وبوفارس صبري. الفريق مسؤول عن تطوير خوارزميات الذكاء الاصطناعي والتحسين، إضافة الى ذلك تم توزيع المهام الجانبية بينهم كما يلي:

1. المهام الجانبية للطالب بوربيع محمد نبيل:

- تصميم واجهة المستخدم وتطوير التطبيق.
- تطوير النماذج التفاعلية وتجربة المستخدم.

- اختبار البرمجيات لضمان جودتها ووظائفها.
- التعاون مع فريق التصميم لضمان توافق الواجهة مع متطلبات المستخدم.
- إدارة تحديثات التطبيق وإصلاح الأخطاء.

2. المهام الجانبية للطالب سعيد فاضل حسن ناجي:

- جمع البيانات اللازمة وإعدادها للتدريب والتعامل مع قواعد البيانات.
- تصميم وتنفيذ استراتيجيات جمع البيانات.
- تنظيف وتحليل البيانات لضمان جودتها وصحتها.
- تطوير وصيانة قاعدة البيانات المستخدمة لتخزين البيانات.
- تنفيذ عمليات تحويل البيانات والمعالجة المسبقة لجعلها جاهزة للتعلم الآلي.

3. المهام الجانبية للطالب بوفارس صبري:

- مسؤول عن الحماية، ويتضمن ذلك ضمان أمان المشروع من الهجمات الإلكترونية وحماية البيانات.
- التسويق والإشهار للمشروع، والبحث عن الشركاء الاقتصاديين والمستثمرين.
- إعداد وتطبيق استراتيجيات تسويقية لزيادة الوعي بالمشروع.
- إدارة الحملات الإعلانية على المنصات المختلفة.
- التواصل مع الشركات والمستثمرين المحتملين لعرض المشروع وجذب الدعم المالي.
- تنظيم الفعاليات والعروض التقديمية لترويج المشروع وزيادة فرص التعاون.
- تحليل ردود الأفعال والتغذية الراجعة من المستخدمين لتحسين وتطوير المشروع.

بهذه الطريقة يتم توزيع المهام بشكل يضمن تكامل الجهود بين أعضاء الفريق لتحقيق أهداف المشروع بنجاح.

4.1 أهداف المشروع

- توفير وسيلة سهلة ومريحة للمستخدمين للوصول إلى معلومات حول وسائل النقل العامة وتخطيط رحلاتهم بكل يسر وسلاسة.
- زيادة الكفاءة في النقل: توجيه المسافرين إلى أفضل الطرق والوسائل للوصول إلى وجهاتهم بأقل وقت وجهد ممكنين.
- تقليل التكاليف: توفير خدمة مجانية أو بتكلفة منخفضة للمستخدمين، مما يساعدهم على توفير النفقات المتعلقة بالنقل.
- الحفاظ على البيئة: تشجيع استخدام وسائل النقل العامة، مما يقلل من ازدحام الطرق وانبعاثات الغازات الضارة.

- التوسع والنمو: توسيع نطاق الخدمة لتشمل المزيد من المدن والمناطق لتعميم الفوائد على أكبر عدد ممكن من الناس.
- تعزيز الاقتصاد المحلي: دعم النشاط الاقتصادي في المدن من خلال تعزيز التجارة والتنمية الحضرية وخلق فرص عمل جديدة في قطاع النقل.
- توفير خدمة شاملة: توفير نظام شامل يغطي جميع وسائل النقل العامة في المدينة، بما في ذلك الحافلات، والمترو، والترام، وغيرها، لتلبية احتياجات جميع شرائح المجتمع.
- تعزيز السلامة: تحسين سلامة الركاب من خلال توفير معلومات دقيقة ومحدثة حول الجداول والطرق الآمنة للوصول إلى المحطات ووسائل النقل.
- تحسين الوصولية: زيادة إمكانية الوصول إلى وسائل النقل العامة لذوي الاحتياجات الخاصة وكبار السن والأشخاص ذوي الدخل المحدود

5.1 الجدول الزمني

1.5.1 :التخطيط والتحضير

- تحديد المتطلبات والأهداف: تحليل السوق وتحديد المتطلبات الوظيفية والتقنية للتطبيق.
- جمع البيانات: جمع البيانات المتعلقة بالطرقات، نقاط التوقف، خطوط النقل، الأسعار، المسافات وكل المعلومات اللازمة لتطوير التطبيق.

2.5.1 التصميم و التطوير

- **تصميم قاعدة بيانات**: تصميم قاعدة بيانات متكاملة و منظمة.
- **تطوير النموذج الأولي**: تطوير النموذج الأولي للتطبيق لتجربة واجهة المستخدم و تدفقات العمل.
- **تطوير البرمجيات**: برمجة التطبيق الأساسي و ربط قاعدة البيانات.
- **تصميم واجهة المستخدم**: تصميم واجهة المستخدم و تجربة المستخدم.
- **تطوير خوارزميات الذكاء الإصطناعي**: تطوير و تدريب خوارزميات الذكاء الإصطناعي لتحسين دقة التوجيهات و التنبؤات.
- **اختبار التطبيق**: اجراء اختبارات شاملة للتطبيق لضمان الأداء السليم و خلوه من الأخطاء.

3.5.1 الإطلاق والتسويق

- **إعداد استراتيجية التسويق**: إعداد و تنفيذ خطة تسويقية شاملة لجذب المستخدمين.
- **إطلاق النسخة التجريبية**: إطلاق نسخة تجريبية (Beta) لجمع التغذية الراجعة من المستخدمين وتحسين التطبيق.
- **إطلاق النسخة النهائية**: إطلاق النسخة النهائية من التطبيق على متاجر التطبيقات.

4.5.1 الدعم والتحسين المستمر

- **دعم العملاء**: توفير دعم فني للمستخدمين و تلقي ملاحظاتهم.
- **التحسينات و التحديثات**: تحديث التطبيق و تحسينه بناءا على ملاحظات المستخدمين و تحليل البيانات.

هذا الجدول الزمني يوفر خريطة طريق واضحة لإطلاق تطبيق النقل العام، ويضمن أن كل مرحلة تحصل على الوقت والموارد اللازمة لضمان النجاح.

المرحلة	المدة الزمنية (الأسابيع)
التخطيط و التحضير	10
التصميم والتطوير	20
الإطلاق و التسويق	8
الدعم والتحسين المستمر	مستمر
إجمالي المدة الزمنية	**38 أسبوع**

جدول 1.1: الجدول الزمني لإطلاق التطبيق

المحور 2 الجوانب الإبتكارية

1.2 طبيعة الإبتكارات

يعرف الابتكار التكنولوجي بأنه عملية إنشاء وتطبيق تقنيات وأدوات وأنظمة وعمليات جديدة أو محسّنة تُحدِث تقدماً كبيراً أو اختراقات في مختلف المجالات. يتضمن هذا الابتكار تسخير المعرفة والخبرة والموارد لتطوير حلول مبتكرة تُعالج المشكلات، وتحسن الكفاءة، وتدفع بالتقدم، وتُضيف قيمة. استناداً إلى هذا التعريف، تمكّنا من تصنيف مشروعنا ضمن الابتكارات التكنولوجية.

2.2 مجالات الإبتكار

تعتبر وسائل النقل العامة حلاً مهماً لتحسين حركة المرور وتقليل ازدحام الطرق، لكنها غالباً ما تتطلب من المستخدمين التفاعل مع الآخرين للحصول على المعلومات اللازمة للتنقل. من هذا المنطلق، يسعى مشروعنا لتحقيق تحول في تجربة استخدام وسائل النقل العامة من خلال تقديم تطبيق يسمح للمستخدمين بمعرفة سبل التنقل بكل سهولة دون الحاجة لطلب المساعدة من الآخرين.

1.2.2 تطبيق تكنولوجيا متقدمة

التقنيات التكنولوجية المتقدمة في هذا المشروع تعتمد على مجموعة من التقنيات الحديثة لتحسين تجربة استخدام وسائل النقل العامة. إليك بعض التفاصيل:

- تطبيق الهاتف المحمول: يتم توفير تطبيق متنقل يعمل على نظامي Android و iOS يتيح للمستخدمين الوصول إلى خدمات النقل العامة بسهولة.
- نظام المعلومات الجغرافية (GIS): يستخدم نظام GIS لتحديد مواقع التوقف ومسارات النقل العامة وتوجيه المستخدمين إليها بطريقة أكثر فعالية.
- تقنيات الذكاء الاصطناعي (AI): يستخدم الذكاء الاصطناعي لتحليل بيانات النقل وتوقع حركة المرور وتوفير اقتراحات لتحسين تجربة المستخدمين.

- تقنيات تعلم الآلة (Machine Learning): يستخدم تقنيات تعلم الآلة لتحليل سلوك المستخدمين وتقديم خدمات مخصصة تلبي احتياجاتهم.

هذه التقنيات تجمع بين التكنولوجيا الحديثة والابتكار لتحسين تجربة استخدام وسائل النقل العامة وجعلها أكثر سلاسة وكفاءة للمستخدمين.

2.2.2 نموذج الأعمال المبتكر

- الاشتراك الشهري/السنوي: يتيح النموذج للمستخدمين الاشتراك في خدمة توفير وسائل النقل العامة بتكلفة ثابتة شهرية أو سنوية.
- العروض والخصومات: يقدم النموذج عروضًا وخصومات للمشتركين الذين يستخدمون الخدمة بانتظام، مما يشجع على استخدام وسائل النقل العامة.
- التعاون مع الشركات: يتيح النموذج التعاون مع الشركات والمؤسسات لتوفير حلول تنقل مخصصة لموظفيها.
- التكامل مع الخدمات الأخرى: يمكن أن يتضمن النموذج تكامل مع خدمات أخرى مثل تأجير السيارات أو خدمات توصيل الطعام أو الخدمات السياحية لتوفير تجربة تنقل شاملة.

هذه النماذج تهدف إلى جعل استخدام وسائل النقل العامة أكثر ملائمة للمستخدمين، وتشجيعهم على الاستفادة الكاملة من الخدمات المقدمة.

3.2.2 العروض والخصومات

- عروض التسجيل الأولى: يمكن تقديم خصومات للمستخدمين الجدد الذين يسجلون في التطبيق لأول مرة، مثل تخفيض على قيمة الإشتراكات أو فترة محدودة للإستخدام المجاني .
- برامج الولاء: يمكن تقديم برنامج ولاء يكافئ المستخدمين الدائمين بنقاط أو خصومات على قيمة الإشتراكات، مما يحفزهم على استخدام الخدمة بانتظام.
- عروض الموسم: يمكن تقديم عروض خاصة بمناسبات معينة مثل العطلات والمواسم السياحية، حيث يمكن تخفيض أسعار الخدمة أو توفير خدمات إضافية مجانية.
- عروض الشراكات: يمكن تعزيز التعاون مع شركات أخرى لتقديم عروض مشتركة توفر خصومات على الرحلات أو خدمات إضافية للمستخدمين.

إن تقديم العروض والخصومات بشكل مناسب وجذاب يمكن أن يسهم بشكل كبير في جذب المستخدمين وزيادة التفاعل مع التطبيق والخدمة المقدمة.

4.2.2 التحليل البياني

- تحليل بيانات الاستخدام: يتم جمع وتحليل بيانات حول استخدام التطبيق والخدمة، مثل عدد المستخدمين النشطين، والمسافات التي يسافرونها، والأوقات التي يستخدمون فيها التطبيق.
- تحليل ردود الفعل: يتم جمع ملاحظات المستخدمين وتحليلها لفهم ما يعجبهم وما لا يعجبهم في التطبيق، ويمكن استخدام هذه الملاحظات لتحسين التطبيق وتلبية احتياجات المستخدمين بشكل أفضل.
- تحليل الاتجاهات: يتم مراقبة الاتجاهات في استخدام وسائل النقل العامة وتحليلها، مثل الطلب على الرحلات في أوقات معينة أو إلى وجهات معينة، مما يمكن من تحسين خدمات النقل المقترحة وتلبية الطلب بشكل أفضل.
- تحليل تجربة المستخدم: يتم تحليل كيفية تفاعل المستخدمين مع التطبيق، ويمكن تحليل عناصر واجهة المستخدم لتحسين تجربة المستخدم وجعلها أكثر سهولة وفعالية.
- تحليل الأداء: يتم مراقبة أداء التطبيق والخدمة بشكل عام، مثل سرعة التحميل واستجابة التطبيق، ويمكن اتخاذ التحسينات اللازمة لتحسين أداء التطبيق.

من خلال التحليل البياني الشامل، يمكن تحسين تجربة المستخدم وتحقيق أقصى استفادة من الخدمة المقدمة.

5.2.2 التكامل الشامل

التكامل الشامل في هذا السياق يعني توفير تجربة تنقل متكاملة للمستخدمين من خلال توفير وسائل نقل عامة مختلفة وتكاملها بشكل سلس مع بعضها البعض ومع التطبيق. إليك بعض النقاط التي تسهم في تحقيق التكامل الشامل:

- تكامل وسائل النقل: يمكن توفير تكامل بين وسائل النقل المختلفة مثل الحافلات والقطارات والمترو، مما يسمح للمستخدمين بالتنقل بسهولة بينها دون الحاجة لعمليات تحويل معقدة.
- تكامل الدفع: يمكن توفير نظام دفع موحد يسمح للمستخدمين بدفع تذاكر النقل العامة والخدمات الأخرى المتعلقة بالتنقل عبر التطبيق، مما يوفر لهم الوقت والجهد.
- تكامل مع الخدمات الأخرى: يمكن تكامل التطبيق مع خدمات أخرى مثل تأجير السيارات أو خدمات توصيل الطعام لتوفير تجربة تنقل شاملة ومريحة للمستخدمين.
- تكامل مع المعلومات الحكومية: يمكن تكامل التطبيق مع قواعد البيانات الحكومية لتوفير معلومات دقيقة حول الجداول الزمنية والمسارات وأماكن التوقف، مما يجعل تجربة المستخدم أكثر سلاسة وفعالية.
- تكامل مع وسائل التواصل الاجتماعي: يمكن تكامل التطبيق مع وسائل التواصل الاجتماعي لتمكين المستخدمين من مشاركة تجاربهم والحصول على توجيهات ونصائح من أصدقائهم ومتابعيهم.

باستخدام التكامل الشامل، يمكن تحسين تجربة المستخدم وجعل التنقل في وسائل النقل العامة أكثر سلاسة وكفاءة.

6.2.2 الاستدامة البيئية

الاستدامة البيئية تعني الحفاظ على البيئة والموارد الطبيعية على المدى الطويل من خلال اتخاذ إجراءات تقلل من الآثار السلبية على البيئة وتعزز الاستدامة البيئية. في سياق تطبيق النقل العام، يمكن تحقيق الاستدامة البيئية من خلال النقاط التالية:

- تشجيع الرحلات المشتركة: تعزيز مفهوم المشاركة في الرحلات لتقليل عدد السيارات على الطريق، مما يقلل من انبعاثات الكربون واستهلاك الطاقة.
- التوعية البيئية: تقديم حملات توعية للمسافرين حول أهمية الاستدامة البيئية وكيفية تقليل الآثار البيئية السلبية لاستخدام وسائل النقل العامة.

تحقيق الاستدامة البيئية في مجال النقل العام يساهم في حماية البيئة وتوفير بيئة صحية ونظيفة للمجتمع.

المحور 3 التحليل الاستراتيجي للسوق

1.3 عرض القطاع السوقي

1.1.3 السوق المحتمل

يتواجد المستخدمون المستهدفون في المدن الكبرى التي تتمتع ببنية تحتية قوية لوسائل النقل العامة. يتواجد هؤلاء في أماكن مثل:

- المناطق الحضرية الكبرى: حيث تعتمد نسبة كبيرة من السكان على وسائل النقل العامة للتنقل.
- المناطق السياحية: التي تجذب الزوار وتحتاج إلى تسهيل عملية التنقل لهم. الجامعات والمدارس: حيث يستخدم الطلاب وسائل النقل العامة بانتظام.
- المناطق الصناعية والتجارية: التي تشهد تنقلاً كبيراً للموظفين بشكل يومي.

2.1.3 أعداد السوق المحتملين

بناءً على تصريحات وزير النقل محمد الحبيب زهانة يوم الثلاثاء الموافق 27 مارس 2023، تم استعراض حصيلة نشاطات قطاع النقل العام لعام 2023، وعلى حسب هذا التصريح يمكننا حساب أعداد السوق المحتمل للمسافرين في الشبكة الداخلية فقط كالتالي:

- **المطارات المفتوحة على الملاحة الجوية العمومية**: 6.7 مليون مسافر.
- **مطار الجزائر**: 2.4 مليون مسافر (هؤلاء مشمولون في الإجمالي السابق 6.7 مليون).
- **شركة الخطوط الجوية الجزائرية**:2.3 مليون مسافر (هؤلاء مشمولون في الإجمالي السابق 6.7 مليون).
- **شركة طيران الطاسيلي**:1.2 مليون مسافر (ملاحظة: هؤلاء مشمولون في الإجمالي السابق 6.7 مليون).
- **النقل بالسكك الحديدية**: 36 مليون مسافر.
- **شركة استغلال الترامواي "سيترام"**: 100 مليون مسافر.

- **مؤسسة "مترو الجزائر"**: 46 مليون مسافر.
- **النقل بالكوابل**: 4.8 مليون مسافر.
- **النقل عبر الطرقات(المسافرين من المؤسسات العمومية للنقل الحضري والشبه الحضري)**: 205 مليون مسافر.

بالتالي، إجمالي عدد المسافرين في الشبكة الداخلية، بافتراض أن جميع الأرقام المذكورة هي مستقلة وليست شاملة لأرقام أخرى، هو مجموع الأعداد التالية:

- المطارات (6.7 مليون مسافر)
- السكك الحديدية (36 مليون مسافر)
- الترامواي (100 مليون مسافر)
- المترو (46 مليون مسافر)
- النقل بالكوابل (4.8 مليون مسافر)
- النقل عبر الطرقات (205 مليون مسافر)

المجموع الكلي: 6.7 + 36 + 100 + 46 + 4.8 + 205 = 402.1 مليون مسافر إذن، أعداد السوق المحتمل للمسافرين في الشبكة الداخلية فقط هو حوالي 402.1 مليون مسافر سنوياً.

3.1.3 السوق المستهدف

يستهدف تطبيقنا الأفراد الذين يعتمدون على وسائل النقل العامة في حياتهم اليومية، ويشمل ذلك الموظفين والطلاب والسياح. هؤلاء المستخدمين يبحثون عن وسائل تضمن لهم تجربة تنقل سلسة وفعالة بدون الحاجة إلى استشارة الآخرين أو الاعتماد على جداول زمنية غير دقيقة.

2.3 التحليل الإستراتيجي للسوق

1.2.3 من سيستخدم تطبيقنا

1. الموظفون: الذين يحتاجون إلى التنقل اليومي للعمل.
2. الطلاب: الذين يستخدمون وسائل النقل العامة للوصول إلى الجامعات والمدارس.
3. السياح: الذين يزورون المدن ويحتاجون إلى إرشادات دقيقة للتنقل بسهولة.
4. كبار السن: الذين قد يواجهون صعوبة في التنقل بمفردهم ويبحثون عن وسيلة بسيطة وآمنة لاستخدام وسائل النقل العامة.

2.2.3 دوافع الاستخدام

1. الدقة والموثوقية: يقدم التطبيق معلومات دقيقة حول مواعيد الحافلات والقطارات، مما يقلل من احتمالية الانتظار الطويل أو فوات الرحلات.
2. التخصيص: يوفر التطبيق إرشادات مخصصة لكل مستخدم بناءً على موقعه ووجهته.
3. التوفير في الوقت: من خلال تحسين مسارات التنقل وتقديم البدائل في حال حدوث تأخير أو عطل.
4. الراحة: يتيح التطبيق للمستخدمين التخطيط المسبق لرحلاتهم وتجنب الاستشارات المتكررة أو الاعتماد على الآخرين.

3.2.3 مبررات اختيار هذا السوق

1. الحاجة الملحة: تزايد الاعتماد على وسائل النقل العامة في المدن الكبرى نتيجة لزيادة التعداد السكاني والتوسع الحضري.
2. التكنولوجيا المتقدمة: تطور تقنيات الذكاء الاصطناعي وسهولة دمجها في التطبيقات لتقديم خدمات مخصصة وفعالة.
3. النمو السكاني: ازدياد عدد السكان في المدن الكبرى واستمرار التوسع العمراني يزيد من الحاجة إلى حلول تنقل مبتكرة.
4. التوجه نحو الاستدامة: تشجيع الحكومات لاستخدام وسائل النقل العامة كجزء من خطط الاستدامة البيئية وتقليل انبعاثات الكربون.

4.2.3 إمكانية إبرام عمليات شراء

يمكن عقد شراكات مع شركات النقل العام لتوفير البيانات اللازمة وتحسين دقة التطبيق. بالإضافة إلى ذلك، يمكن إبرام اتفاقيات مع مزودي خدمات الإعلان للترويج للعروض الخاصة والنشاطات المحلية التي تستهدف مستخدمي النقل العام. كما يمكن توفير نسخ مدفوعة من التطبيق تحتوي على ميزات إضافية، مثل التنبيهات الفورية والتوجيهات الحية، مما يفتح مجالاً لتحقيق الإيرادات المستدامة.

3.3 قياس شدة المنافسة في السوق الجزائرية

1.3.3 المنافسين المباشرين

1. تطبيق MyBus: هو تطبيق مخصص للطلبة الجامعيين لمعرفة خطوط النقل الجامعي. يتميز ب تركيز مخصص على الطلاب الجامعيين ويوفر معلومات دقيقة ومحدثة حول خطوط النقل الجامعي أما نقاط الضعف فهي محدودية

الجمهور المستهدف (الطلبة الجامعيين فقط)، قد يفتقر إلى البيانات المتعلقة بوسائل النقل الأخرى. أما حصته السوقية فيقدر أن يكون لديه حصة سوقية محدودة بحكم تخصصه في شريحة معينة من المستخدمين (الطلبة الجامعيين).

2. تطبيقات شركة النقل ETUS: تطبيقات مخصصة لحافلات شركة النقل ETUS. يتميز بتوفير معلومات دقيقة حول جداول حافلات شركة النقل، موثوقية البيانات الصادرة من مصدر رسمي. أما نقاط الضعف فهي تقتصر على حافلات شركة ETUS فقط، قد تفتقر إلى تكامل مع وسائل نقل الأخرى، وأما بالنسبة للحصة السوقية فإنه يقدر أن يكون لديه حصة سوقية جيدة بين مستخدمي حافلات شركة النقل ETUS ولكن قد يكون محدوداً في التغطية الجغرافية حسب نطاق عمل الشركة.

3. تطبيق Mahatati: تطبيق تملكه الشركة الوطنية لاستغلال المحطات البرية في الجزائر، يقدم خدمة السفر بين الولايات مع إمكانية حجز التذاكر ومعرفة مواعيد انطلاق الحافلات. حيث يتميز بتقديم خدمات شاملة للسفر بين الولايات بما في ذلك حجز التذاكر، موثوقية البيانات من مصدر رسمي وتغطية واسعة تشمل محطات النقل بين الولايات. أما نقاط الضعف فهو يركز على السفر بين الولايات فقط، قد لا تغطي التنقل داخل المدن، تفتقر إلى بعض الميزات التكنولوجية المتقدمة

2.3.3 المنافسين الغير مباشرين

1. Google: Maps يتميز بقاعدة بيانات ضخمة وشاملة وتكامل مع العديد من الخدمات الأخرى. أما نقاط الضعف عدم التركيز بشكل كامل على النقل العام، بيانات قد لا تكون دقيقة دائماً بالنسبة للجزائر والحصة السوقية حيث يستخدم بشكل كبير.

2. شركات نقل السيارات(Yessir, Hitch, Yango...): تطبيقات جزائرية توفر خدمات التنقل عبر السيارات الخاصة. تتميز بخدمة محلية تلبي احتياجات المستخدمين الجزائريين وسهولة الاستخدام وخيارات الدفع المتعددة. لكن لها محدودية في التغطية خارج المدن الكبرى وتكلفة أعلى مقارنة بوسائل النقل العام.

تشهد السوق الجزائرية لتطبيقات النقل العام منافسة من تطبيقات متخصصة مثل MyBus، وتطبيقات شركة النقل ETUS، وMahatati، بالإضافة إلى شركات نقل السيارات مثل Yassir، Hitch وYango. توفر هذه التطبيقات معلومات دقيقة ولكنها محدودة من حيث التغطية والنطاق. يتميز تطبيقنا المقترح من خلال تقديم تغطية شاملة لجميع وسائل النقل العام، والاستفادة من تقنيات الذكاء الاصطناعي لتقديم إرشادات مخصصة لكل مستخدم، مما يمكنه من تلبية احتياجات مجموعة أوسع من المستخدمين وتحقيق ميزة تنافسية في السوق.

4.3 الإستراتيجية التسويقية

1.4.3 تحليل السوق والفهم العميق للزبائن

1. **دراسة السوق المستهدفة:**

- الفئة المستهدفة: الطلاب الجامعيين، العاملين، السياح، وكل من يعتمد على وسائل النقل العام.
- احتياجاتهم: معلومات دقيقة ومحدثة، سهولة الوصول، التوجيهات المخصصة، خيارات الدفع المتعددة.

2. **جمع البيانات وتحليلها**:

- استخدام الاستطلاعات والدراسات الميدانية لفهم عادات وتفضيلات المستخدمين.
- تحليل بيانات الاستخدام لتحديد الأنماط والاحتياجات غير الملباة.

2.4.3 تحديد الميزة التنافسية

1. **التقنيات المستخدمة**:

- استخدام الذكاء الاصطناعي لتقديم معلومات دقيقة وتوجيهات مخصصة.
- تقديم خدمات متكاملة تشمل جميع وسائل النقل العام (الحافلات، القطارات، الترام، وما إلى ذلك).

2. **العروض الفريدة**:

- تقديم خدمات حجز التذاكر إلكترونيًا.
- توفير خرائط تفاعلية وجداول زمنية محدثة.
- تقديم إخطارات فورية عن أي تغييرات أو تأخيرات.

3.4.3 استخدام قنوات التسويق الفعالة

1. **التسويق الرقمي**:

- **الإعلانات على وسائل التواصل الاجتماعي**: الاستفادة من الفيسبوك، الإنستغرام، تويتر، ولينكد إن لاستهداف الجمهور المناسب.
- **إعلانات البحث المدفوعة**: استخدام إعلانات جوجل لتحسين ظهور التطبيق عند البحث عن خدمات النقل.
- **التسويق عبر البريد الإلكتروني**: إرسال رسائل تسويقية مخصصة للمستخدمين المسجلين.

2. **التسويق عبر الهواتف المحمولة**:

- **الإشعارات الفورية**: إرسال إشعارات فورية للمستخدمين عن التحديثات والعروض الجديدة.
- **الإعلانات داخل التطبيقات الأخرى**: التعاون مع تطبيقات أخرى لعرض الإعلانات داخل تطبيقاتهم.

3. **التسويق التقليدي**:

- الإعلانات في الأماكن العامة: نشر الإعلانات في المحطات، الحافلات، والترام.
- العروض الترويجية: تقديم خصومات وعروض مجانية لفترة محددة لتشجيع الاستخدام الأولي.

4.4.3 تحسين تجربة المستخدم

1. **تصميم واجهة مستخدم سهلة الاستخدام:**

- تصميم واجهة مستخدم بسيطة وسهلة الفهم.
- تحسين تجربة المستخدم بناءً على ملاحظاتهم.

2. **خدمة العملاء:**

- توفير دعم فني على مدار الساعة عبر الهاتف، البريد الإلكتروني والدردشة المباشرة.
- تقديم مركز مساعدة شامل على التطبيق يحتوي على إجابات للأسئلة الشائعة.

5.4.3 استراتيجيات الاحتفاظ بالعملاء

1. **البرامج الولائية**: المكافآت والخصومات على الاشتراكات الشهرية أو السنوية.
2. **التحديثات الدورية**: إضافة ميزات جديدة وتحسينات مستمرة للتطبيق بناءً على ملاحظات المستخدمين.
3. **التفاعل المستمر مع العملاء**: التفاعل مع المستخدمين عبر وسائل التواصل الاجتماعي والاستجابة لتعليقاتهم واقتراحاتهم.

6.4.3 الشراكات الاستراتيجية

1. **التعاون مع شركات النقل المحلية:**

- عقد شراكات مع شركات النقل العام والخاصة لتعزيز التكامل بين خدمات النقل.
- التفاوض على تقديم خصومات للمستخدمين الذين يشترون الباقات عبر التطبيق.

2. **التعاون مع المؤسسات التعليمية والشركات**: عقد اتفاقيات مع الجامعات والشركات لتقديم خدمات مخصصة للطلاب والموظفين.

7.4.3 تقييم الأداء والتحسين المستمر

1. **مراقبة الأداء:**

- استخدام أدوات التحليل لمراقبة أداء الحملات التسويقية وقياس عائد الاستثمار.
- تحليل بيانات الاستخدام لتحديد نقاط القوة والضعف.

2. **التكيف والتحسين**: تعديل الاستراتيجيات بناءً على نتائج التحليل والملاحظات المستمرة من المستخدمين.

المحور 4 خطة الإنتاج والتنظيم

1.4 خطة الإنتاج

1.1.4 التخطيط والتحضير

1. **تحديد المتطلبات والأهداف**

(ا) **اجتماعات تعريفية**: عقد اجتماعات مع الفرق المعنية (التطوير، التسويق، الإدارة) لتحديد المتطلبات والأهداف.

(ب) **تحليل السوق**: إجراء بحث شامل لفهم احتياجات السوق المستهدف وتحديد الميزات الأساسية التي يجب أن يوفرها التطبيق.

(ج) **تحديد الخصائص**: وضع قائمة مفصلة بالخصائص والمميزات المطلوبة للتطبيق.

1. **جمع البيانات:**

(ا) **تحديد المصادر**: تحديد مصادر البيانات اللازمة مثل السلطات المحلية، شركات النقل، المواقع الإلكترونية، وبيانات GPS.

(ب) **جمع البيانات الميدانية**:إرسال فرق ميدانية لجمع بيانات الطرقات ونقاط التوقف.

(ج) **تنظيم البيانات**: تنظيم البيانات في ملفات وقواعد بيانات مناسبة للاستخدام لاحقًا.

2.1.4 التصميم والتطوير

1. **تصميم قاعدة البيانات:**

(ا) **هيكلة قاعدة البيانات**: تصميم هيكل قاعدة البيانات بما يتناسب مع نوعية البيانات التي تم جمعها.

(ب) **إنشاء قاعدة البيانات**: استخدام أدوات إدارة قواعد البيانات لإنشاء قاعدة البيانات.

1. **تطوير النموذج الأولي:**

(ا) **تصميم الواجهات الأولية**: تصميم واجهات المستخدم الأولية وتدفقات العمل.

(ب) **اختبار النموذج الأولي**: اختبار النموذج الأولي داخليًا لجمع الملاحظات وتحسين التصميم.

1. **تطوير البرمجيات:**

(ا) **برمجة التطبيق**: البدء ببرمجة التطبيق باستخدام لغات البرمجة المناسبة مثل Java، Kotlin، Swift.

(ب) **دمج قاعدة البيانات**: ربط التطبيق بقاعدة البيانات لضمان تدفق البيانات بشكل صحيح.

(ج) **اختبارات الوحدات**: إجراء اختبارات أولية على وحدات التطبيق للتأكد من عملها بشكل صحيح.

1. **تصميم واجهة المستخدم:**

(ا) **تطوير واجهة المستخدم**: تطوير واجهة مستخدم جذابة وسهلة الاستخدام.

(ب) **تجربة المستخدم**: تحسين تجربة المستخدم من خلال إجراء اختبارات واستخدام التعليقات لتحسين التصميم.

1. **تطوير خوارزميات الذكاء الإصطناعي:**

(ا) **تطوير الخوارزميات**: تطوير خوارزميات الذكاء الاصطناعي لتحليل البيانات وتقديم التوجيهات.

(ب) **تدريب النماذج**: تدريب نماذج الذكاء الاصطناعي باستخدام البيانات المجمعة.

(ج) **اختبار النماذج**: اختبار النماذج لضمان دقتها وفعاليتها.

1. **اختبار التطبيق:**

(ا) **اختبار النظام بالكامل**: إجراء اختبارات شاملة للتطبيق للتحقق من أدائه واستقراره.

(ب) **اختبار القبول**: إجراء اختبار قبول المستخدم للتأكد من أن التطبيق يلبي المتطلبات والاحتياجات.

3.1.4 الإطلاق والتسويق

1. **إعداد استراتيجية التسويق:**

- **وضع خطة تسويقية**: إعداد خطة تسويقية تتضمن الإعلان عبر الإنترنت، وسائل التواصل الاجتماعي، والتسويق التقليدي.
- **تصميم المواد التسويقية**: تصميم المواد الترويجية مثل الإعلانات، الفيديوهات التوضيحية، والملصقات.

2. **إطلاق النسخة التجريبية:**

- **إطلاق بيتا**: إطلاق نسخة تجريبية من التطبيق لجمع التغذية الراجعة من المستخدمين.
- **جمع الملاحظات**: جمع وتحليل ملاحظات المستخدمين لتحسين التطبيق.

3. **إطلاق النسخة النهائية:**

- **تحسين التطبيق**: تنفيذ التحسينات النهائية بناءً على ملاحظات النسخة التجريبية.
- **النشر على المتاجر**: نشر التطبيق على متاجر التطبيقات مثل Google Play وApp Store.
- **إطلاق حملة تسويقية**: تنفيذ حملة تسويقية شاملة للترويج للتطبيق وجذب المستخدمين.

4.1.4 الدعم والتحسين المستمر

1. **دعم العملاء:**

- **إنشاء فريق دعم**: إنشاء فريق دعم فني للرد على استفسارات المستخدمين وحل مشاكلهم.
- **تقديم الدعم المستمر**: تقديم الدعم الفني عبر الهاتف، البريد الإلكتروني، والدردشة المباشرة.

2. **التحسينات و التحديثات:**

- **جمع الملاحظات الدورية**: الاستمرار في جمع ملاحظات المستخدمين بشكل دوري.
- **تحديث التطبيق**: إطلاق تحديثات دورية للتطبيق لإضافة ميزات جديدة وتحسين الأداء.
- **تحليل الأداء**: تحليل أداء التطبيق باستمرار واستخدام البيانات لتحسين وتطوير التطبيق.

5.1.4 التقييم والاستدامة

1. **تقييم الآداء:**

- **مراجعة دورية**: مراجعة أداء التطبيق بشكل دوري باستخدام مؤشرات الأداء الرئيسية (KPIs).
- **تحليل العوائد**: تحليل العوائد المالية ورضا المستخدمين لتحديد مدى نجاح التطبيق.

2. **استراتيجيات الإستدامة:**

- **تطوير الميزات المستقبلية**: تحديد وتطوير الميزات المستقبلية بناءً على اتجاهات السوق واحتياجات المستخدمين.
- **توسيع الشراكات**: إقامة شراكات جديدة لتعزيز قاعدة المستخدمين وتحسين الخدمة المقدمة.

2.4 الجدول الزمني

المرحلة	النشاط	المدة الزمنية (الأسابيع)
التخطيط و التحضير	تحديد المتطلبات و جمع البيانات	10
التصميم والتطوير	تصميم قاعدة البيانات، تطوير النموذج الأولي، برمجة التطبيق، تصميم واجهة المستخدم، تطوير خوارزميات الذكاء الإصطناعي، إختبار التطبيق	20
الإطلاق و التسويق	إعداد استراتيجية التسويق، إطلاق النسخة التجريبية، إطلاق النسخة النهائية، تنفيد حملة تسويقية	8
الدعم والتحسين المستمر	دعم العملاء، جمع الملاحظات، التحديثات الدورية	مستمر
التقييم و الإستدامة	تقييم الآداء، استراتيجيات الإستدامة	مستمر
إجمالي المدة الزمنية	//	**38 أسبوع**

جدول 1.4: الجدول الزمني التفصيلي لإطلاق التطبيق

المحور 5 الخطة التسويقية

1.5 الخطة المالية للمشروع ونموذج العمل التجاري

1.1.5 التكاليف والرسوم

سيدعم المشروع عدة تكاليف ورسوم على النحو التالي:

1. تكاليف لوجيستية

- يقدر إيجار المبنى بـ 40,000.00 دج/ شهر.....480,000.00/سنة
- رواتب الموظفين 120,000.00 دج/شهر.. 1,440,000.00/سنة.
- تطوير البرمجيات والتصميم: يشمل تكلفة تطوير التطبيق بما يتضمن واجهة المستخدم والتجربة الجديدة، ويتضمن أيضًا تصميم الرسومات والرموز والصور المستخدمة تقدر ب 800,000.00دج.

2. تكاليف التسويق والدعاية:

- حملات التسويق الرقمي: تكلفة الإعلانات عبر الإنترنت ومواقع التواصل الاجتماعي والترويج للتطبيق. يتوقف تقدير هذه التكاليف على نطاق وحجم الحملة ومدى التغطية المطلوبة. تقدر ب 200,000.00دج.
- إعداد المواد الترويجية: تكلفة إنشاء المواد الترويجية مثل الفيديوهات التعريفية والمطبوعات الدعائية. تقدر ب 20,000.00دج.
- العلاقات العامة: تكلفة الترويج والتسويق من خلال التعاون مع وسائل الإعلام والنشرات الصحفية والفعاليات العامة. تقدر ب150,000.00دج.

3. تكاليف التشغيل و الصيانة:

- شراء وصيانة الأجهزة والمعدات: تكلفة شراء الأجهزة اللازمة لتشغيل التطبيق، أجهزة الكمبيوتر، بالإضافة إلى تكاليف الصيانة الدورية. تقدر ب 500,000.00 دج.

4. تكاليف الإدارة:

- تجهيزات مكتبية 400,000.00 دج.
- التشغيل اليومي للمقر: تكاليف الكهرباء والمياه لمكتب الشركة بالإضافة إلى تكاليف الإنترنت والهواتف واللوازم الأخرى. تقدر تكلفة هذا البند 450,000.00 دج للعام.

5. تكاليف قانونية:

- الإستشارات القانونية: تكلفة الحصول على إستشارات قانونة للمساعدة في إعداد الإتفاقيات و معالجة القضايا القانونية المحتملة. تقدر ب 50,000.00 دج للعام.

2.5 تمويل المشروع

سيتم تمويل المشروع من قبل صندوق الشركات الناشئة الجزائري

سداد الاعتمادات والقروض سيتم تنفيدها على المدى الطويل والذي سيتم التفاوض عليه مع المنظمة المعنية.

الشرط	التكلفة
تطوير التطبيق الاولي	250,000.00 دج
تطوير التطبيق النهائي ووضعه على متجر بلاي ستور وأب ستور	800,000.00 دج
تكاليف التسويق والدعاية	370,000.00 دج
معدات المكتبة	400,000.00 دج
اجهزة الكمبيوتر والبرمجيات والمواقع	500,000.00 دج
أعباء صندوق التأمين CNAS	201,600.00 دج/سنة
تأمين المعدات	90,000.00 دج/سنة
تكاليف الإدارية	50,000.00 دج
رواتب 120,000.00 دج/شهر	1,440,000.00 دج/سنة
ايجار مقر العمل 40,000.00 دج/شهر	480,000.00 دج/سنة
الرسوم الخارجية: المياه، الغاز، الكهرباء، الهاتف، صيانة المعدات، الاتصالات وغيرها	450,000.00 دج/سنة
محاسبون قانونيون ومحامون	82,000.00 دج
المجموع	**5,113,600.00** دج

جدول 1.5: جدول التكاليف والإستثمار (التوقعات)

3.5 رقم الأعمال

1.3.5 النظرة التفاؤلية

- الجدول يوضح تطور أداء الشركة عبر فترات زمنية متعددة، حيث يتناول عدد الاشتراكات، سعر الاشتراك، والإيرادات الإجمالية. يبرز الجدول استقرار سعر الاشتراك على مدى الفترات المحددة، وتقلبات طفيفة في عدد

الاشتراكات عبر الأعوام. هذه البيانات توضح أهمية تحقيق توازن في قاعدة العملاء لضمان استدامة الإيرادات دون التفاوت الكبير في الأداء المالي:

العام	عدد الاشتراكات	سعر الاشتراك	الإيرادات الاجمالية
N	12250	800 دج	9,800.000 دج
N+1	13200	800 دج	10,560.000 دج
N+2	14000	800 دج	11,200.000 دج
N+3	16000	800 دج	12,800.000 دج
N+4	18000	800 دج	14,400.000 دج

جدول 2.5: جدول تفاصيل رقم الأعمال في الحالة المتفائلة

- يبين الجدول الآتي مجموع التكاليف والمبيعات مع هامش الربح:

التكاليف	المبيعات	الربح
25,500.000 دج	48,060.000 دج	22,560.000 دج

جدول 3.5: مجموع التكاليف والمبيعات مع هامش الربح

2.3.5 النظرة التشاؤمية

العام	عدد الاشتراكات	سعر الاشتراك	الإيرادات الاجمالية
N	6300	800 دج	5,040.000 دج
N+1	6320	800 دج	4,960.000 دج
N+2	6300	800 دج	5,040.000 دج
N+3	6400	800 دج	5,120.000 دج
N+4	6300	800 دج	5,040.000 دج

جدول 4.5: جدول تفاصيل رقم الأعمال في الحالة المتشائمة

- يبين الجدول الآتي مجموع التكاليف والمبيعات مع هامش الخسارة:

التكاليف	المبيعات	الخسارة
25,500.000 دج	25,200.000 دج	300.000 دج

جدول 5.5: مجموع التكاليف والمبيعات مع هامش الخسارة

4.5 جداول حسابات النتائج

الجداول الموالية هي حسابات متوقعة لنشاط المشروع في السنوات المقبلة.

الأشهر /المصاريف	1	2	3	4	5	6
تطوير التطبيق	57,500	57,500	57,500	57,500	57,500	57,500
تكاليف التسويق والدعاية	30,000	30,000	30,000	30,000	30,000	30,000
معدات المكتبة	400,000	/	/	/	/	/
البرمجيات والمواقع	41,666	41,666	41,666	41,666	41,666	41,666
أعباء صندوق التأمين CNAS	16,800	16,800	16,800	16,800	16,800	16,800
تأمين المعدات	7,500	7,500	7,500	7,500	7,500	7,500
تكاليف الإدارية	4,166	4,166	4,166	4,166	4,166	4,166
رواتب العمال	120,000	120,000	120,000	120,000	120,000	120,000
تكلفة الايجار	40,000	40,000	40,000	40,000	40,000	40,000
الرسوم الخارجية	37,500	37,500	37,500	37,500	37,500	37,500
محاسبون قانونيون	6,830	6,830	6,830	6,830	6,830	6,830
مجموع النفقات	456,765	456,765	456,765	456,765	456,765	456,765

جدول 6.5: جدول المصاريف (الجزء الأول)

الأشهر /المصاريف	7	8	9	10	11	12	المجموع
تطوير التطبيق	57,500	57,500	57,500	57,500	57,500	57,500	690,000
التسويق والدعاية	30,000	30,000	30,000	30,000	30,000	30,000	360,000
معدات المكتبة	/	/	/	/	/	/	400,000
البرمجيات والمواقع	41,666	41,666	41,666	41,666	41,666	41,666	500,000
صندوق التأمين CNAS	16,800	16,800	16,800	16,800	16,800	16,800	201,600
تأمين المعدات	7,500	7,500	7,500	7,500	7,500	7,500	90,000
تكاليف الإدارية	4,166	4,166	4,166	4,166	4,166	4,166	50,000
رواتب العمال	120,000	120,000	120,000	120,000	120,000	120,000	1,440,000
تكلفة الايجار	40,000	40,000	40,000	40,000	40,000	40,000	480,000
الرسوم الخارجية	37,500	37,500	37,500	37,500	37,500	37,500	450,000
محاسبون قانونيون	6,830	6,830	6,830	6,830	6,830	6,830	82,000
مجموع النفقات	456,765	456,765	456,765	456,765	456,765	456,765	5,113,600

جدول 7.5: جدول المصاريف (الجزء الثاني)

5.5 خطة الخزينة

خطة الخزينة التي سينتهجها مشروعنا هي كالتالي في السنوات المقبلة موضحة في الجداول الموالية:

الأشهر	1	2	3	4	5	6
البيع	683,000	683,000	683,000	683,000	683,000	683,000
صافي المبيعات	683,000	683,000	683,000	683,000	683,000	683,000
مصاريف	456,765	456,765	456,765	456,765	456,765	456,765
هامش الربح	226,235	226,235	226,235	226,235	226,235	226,235
الربح التشغيلي	226,235	226,235	226,235	226,235	226,235	226,235
إيرادات الفوائد	0	0	0	0	0	0
الربح قبل الضريبة	226,235	226,235	226,235	226,235	226,235	226,235
رسوم الضريبة	0	0	0	0	0	0
صافي الإيرادات	226,235	226,235	226,235	226,235	226,235	226,235

جدول 8.5: الإيرادات والنفقات المتوقعة خلال السنة الأولى حسب كل شهر لنشاطنا (الجزء الأول)

الأشهر	7	8	9	10	11	12	المجموع
البيع	683,000	683,000	683,000	683,000	683,000	683,000	8,200,000
صافي المبيعات	683,000	683,000	683,000	683,000	683,000	683,000	8,200,000
مصاريف	456,765	456,765	456,765	456,765	456,765	456,765	5,481,180
هامش الربح	226,235	226,235	226,235	226,235	226,235	226,235	2,714,820
الربح التشغيلي	226,235	226,235	226,235	226,235	226,235	226,235	2,714,820
إيرادات الفوائد	0	0	0	0	0	0	0
الربح قبل الضريبة	226,235	226,235	226,235	226,235	226,235	226,235	2,714,820
رسوم الضريبة	0	0	0	0	0	0	0
صافي الإيرادات	226,235	226,235	226,235	226,235	226,235	226,235	2,714,820

جدول 9.5: الإيرادات والنفقات المتوقعة خلال السنة الأولى حسب كل شهر لنشاطنا (الجزء الثاني)

الأشهر	1	2	3	4	5	6
البيع	1,083,333	1,083,333	1,083,333	1,083,333	1,083,333	1,083,333
صافي المبيعات	1,083,333	1,083,333	1,083,333	1,083,333	1,083,333	1,083,333
مصاريف	516,666	516,666	516,666	516,666	516,666	516,666
هامش الربح	566,666	566,666	566,666	566,666	566,666	566,666
الربح التشغيلي	566,666	566,666	566,666	566,666	566,666	566,666
إيرادات الفوائد	0	0	0	0	0	0
الربح قبل الضريبة	566,666	566,666	566,666	566,666	566,666	566,666
رسوم الضريبة	0	0	0	0	0	0
صافي الإيرادات	566,666	566,666	566,666	566,666	566,666	566,666

جدول 10.5: الإيرادات والنفقات المتوقعة بعد خمس سنوات حسب كل شهر لنشاطنا (الجزء الأول)

الأشهر	7	8	9	10	11	12	المجموع
البيع	1,083,333	1,083,333	1,083,333	1,083,333	1,083,333	1,083,333	13,000,000
صافي المبيعات	1,083,333	1,083,333	1,083,333	1,083,333	1,083,333	1,083,333	13,000,000
مصاريف	516,666	516,666	516,666	516,666	516,666	516,666	000.200.6
هامش الربح	566,666	566,666	566,666	566,666	566,666	566,666	6,800,000
الربح التشغيلي	566,666	566,666	566,666	566,666	566,666	566,666	6,800,000
إيرادات الفوائد	0	0	0	0	0	0	0
الربح قبل الضريبة	566,666	566,666	566,666	566,666	566,666	566,666	6,800,000
رسوم الضريبة	0	0	0	0	0	0	0
صافي الإيرادات	566,666	566,666	566,666	566,666	566,666	566,666	6,800,000

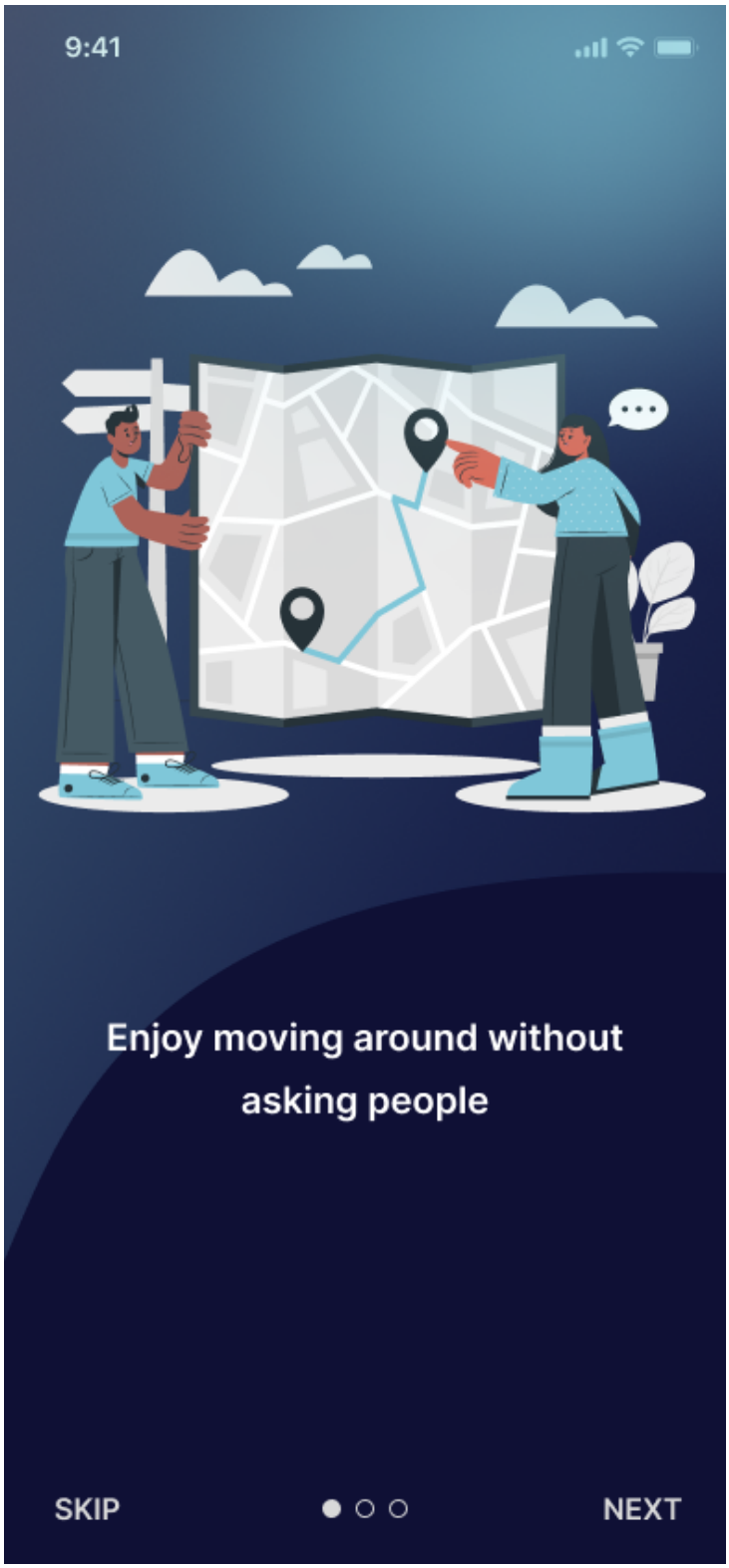
جدول 11.5: الإيرادات والنفقات المتوقعة بعد خمس سنوات حسب كل شهر لنشاطنا (الجزء الثاني)

المحور 6 النموذج التجريبي الأولي

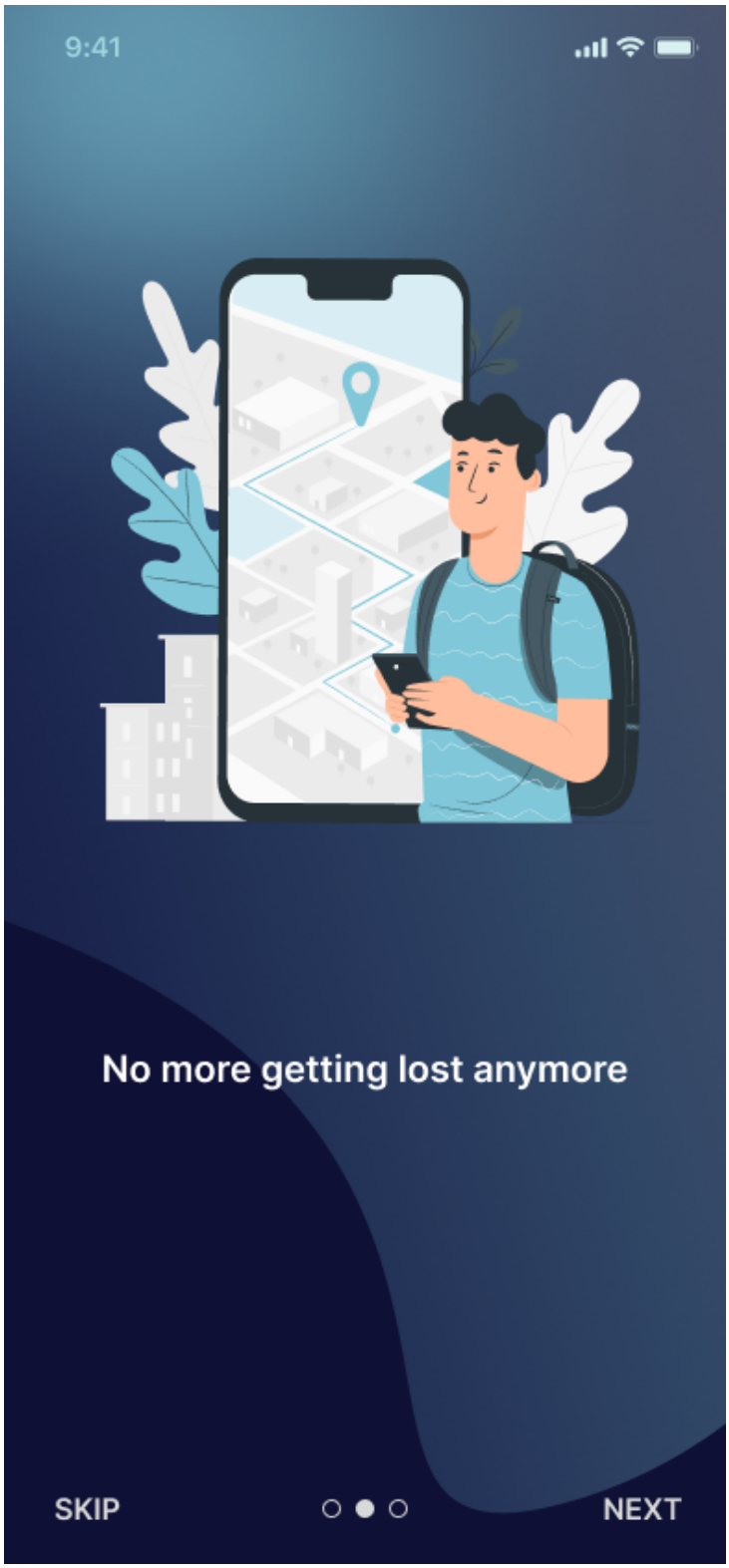
1.6 واجهة المستخدم

في ما يلي، سنعرض بعض لقطات الشاشة لواجهات المستخدم التي تم تطويرها لتطبيقنا.

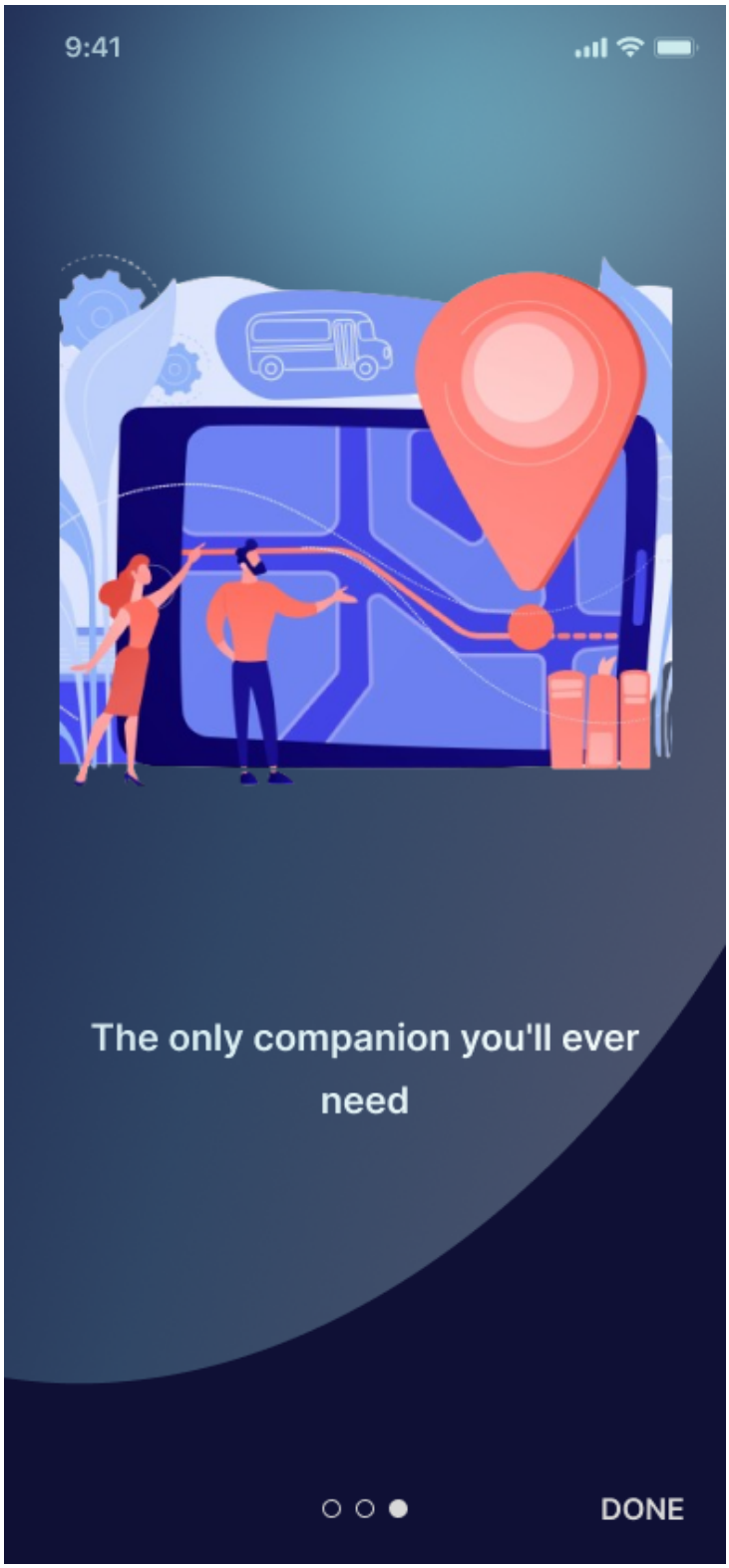
2.6 صفحة البدء



شكل 1.6: واجهة البدء 1



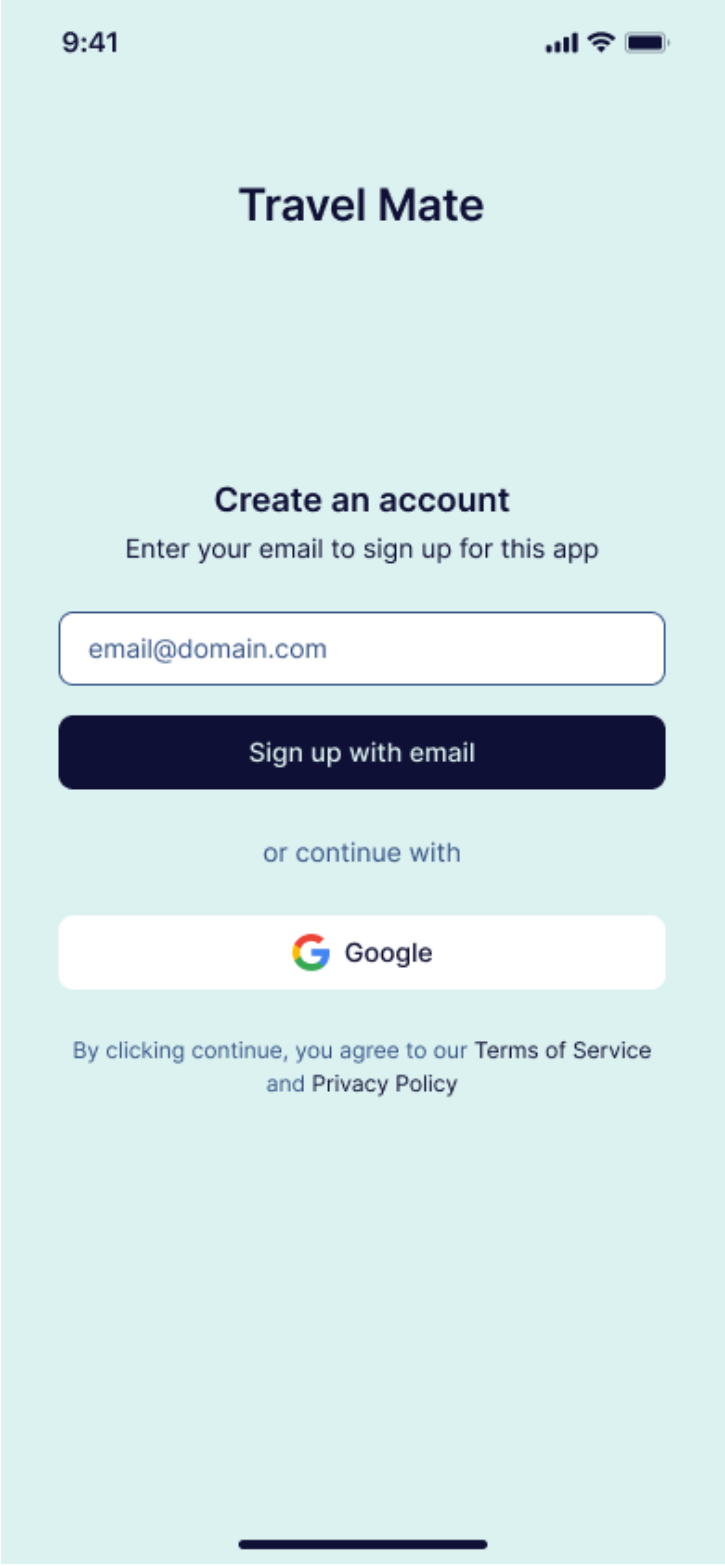
شكل 2.6: واجهة البدء 2



شكل 3.6: واجهة البدء 3

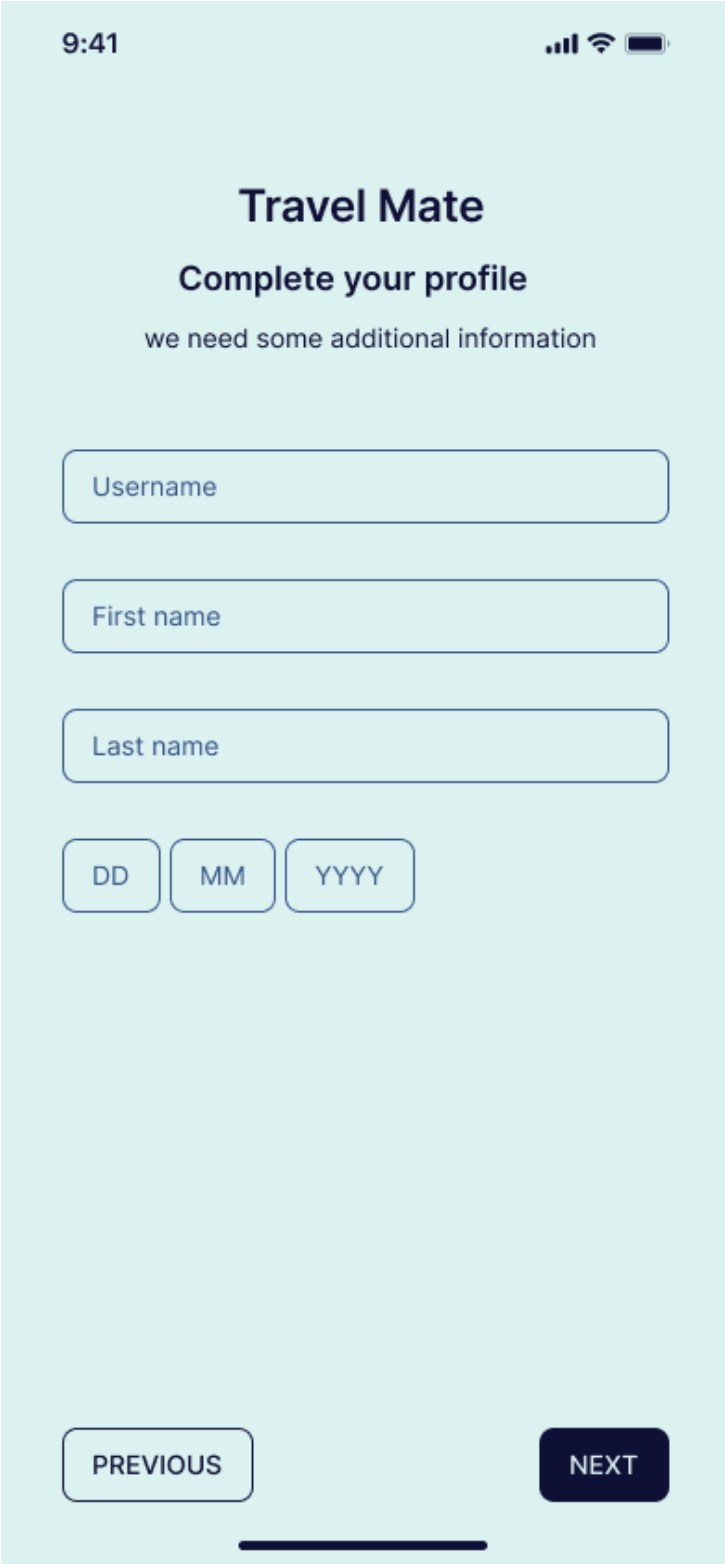
تظهر هذه الصور الثلاثة عند فتح التطبيق لأول مرة، وتوفر برنامجًا تعليميًا يبرز فوائد استخدام تطبيقنا.

3.6 صفحة المصادقة



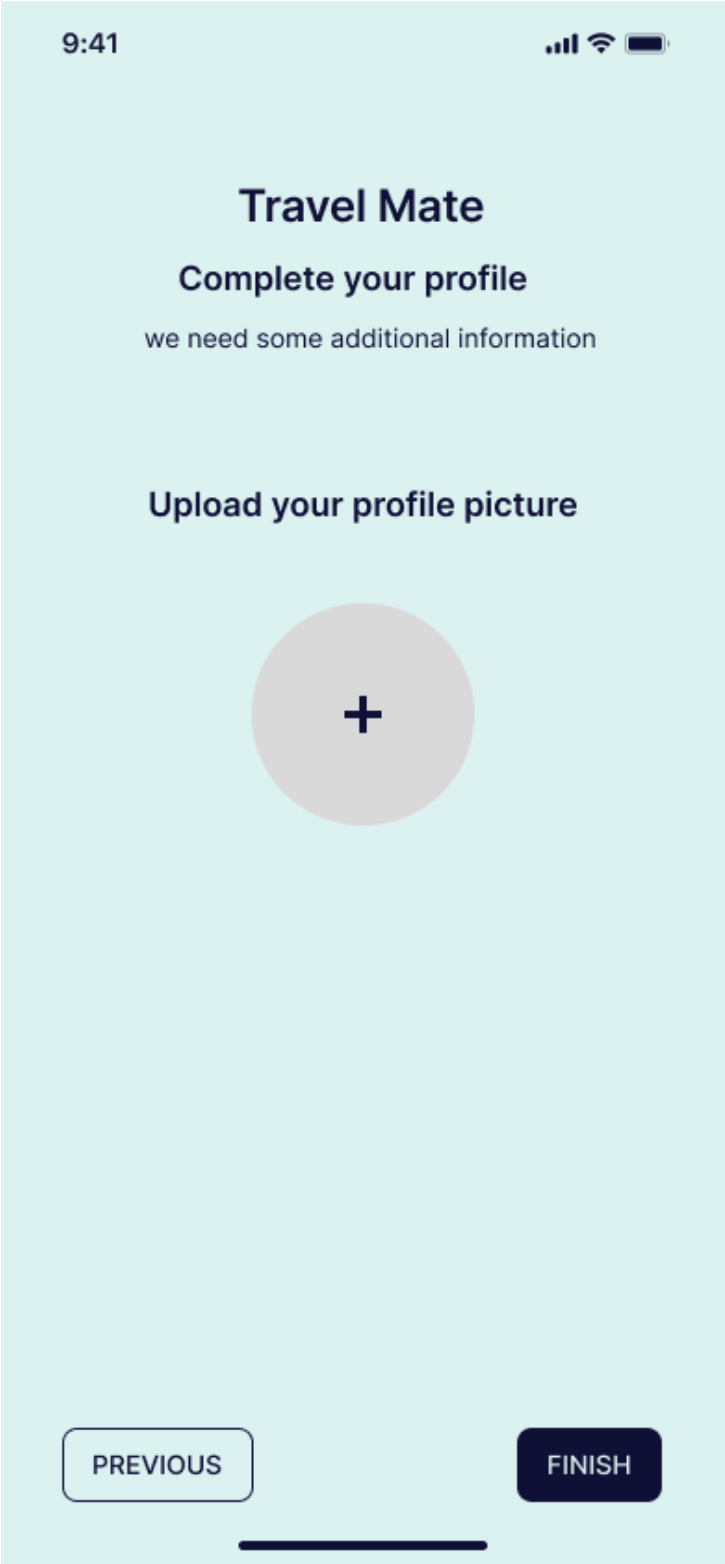
شكل 4.6: واجهة إنشاء الحساب

تصور هذه الصورة صفحة إنشاء الحساب، وتعرض طرقًا مختلفة لإنشاء الحساب.



شكل 5.6: واجهة خطوة تسجيل المعلومات

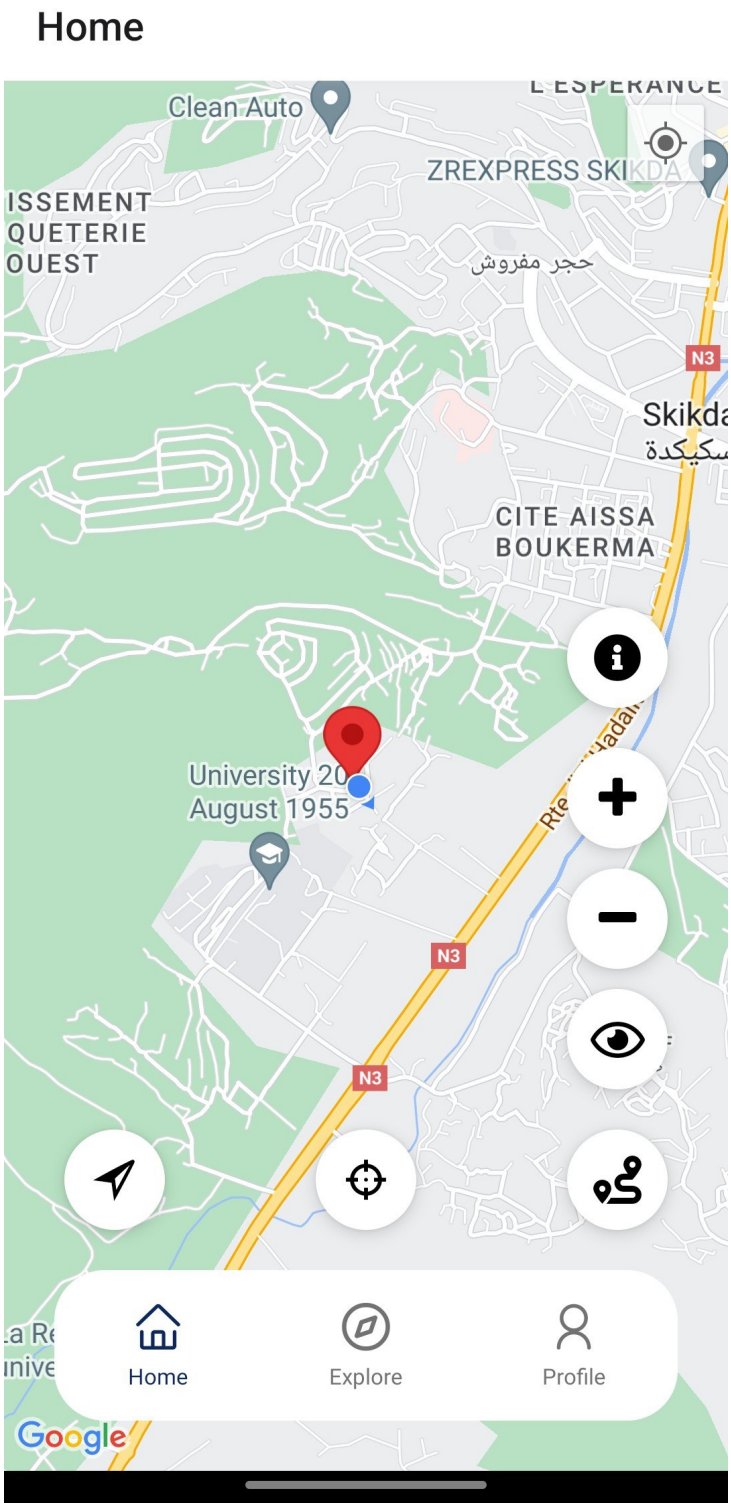
تعرض هذه الصورة صفحة المعلومات الشخصية لإكمال إنشاء الحساب.



شكل 6.6: واجهة خطوة إضافة صورة الحساب

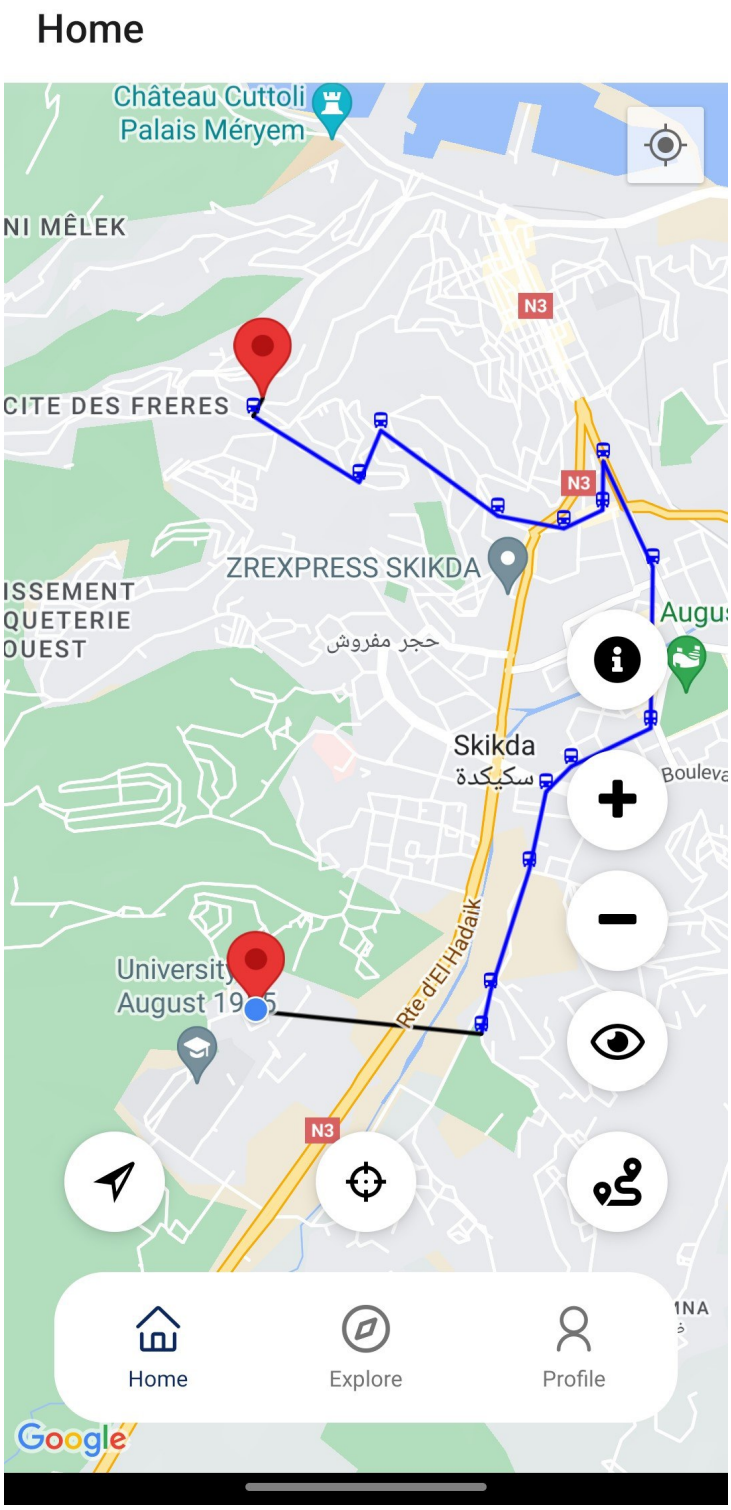
تعرض هذه الصورة قسم إنشاء الملف الشخصي حيث يمكنك تحميل صورة ملفك الشخصي.

4.6 الصفحة الرئيسية



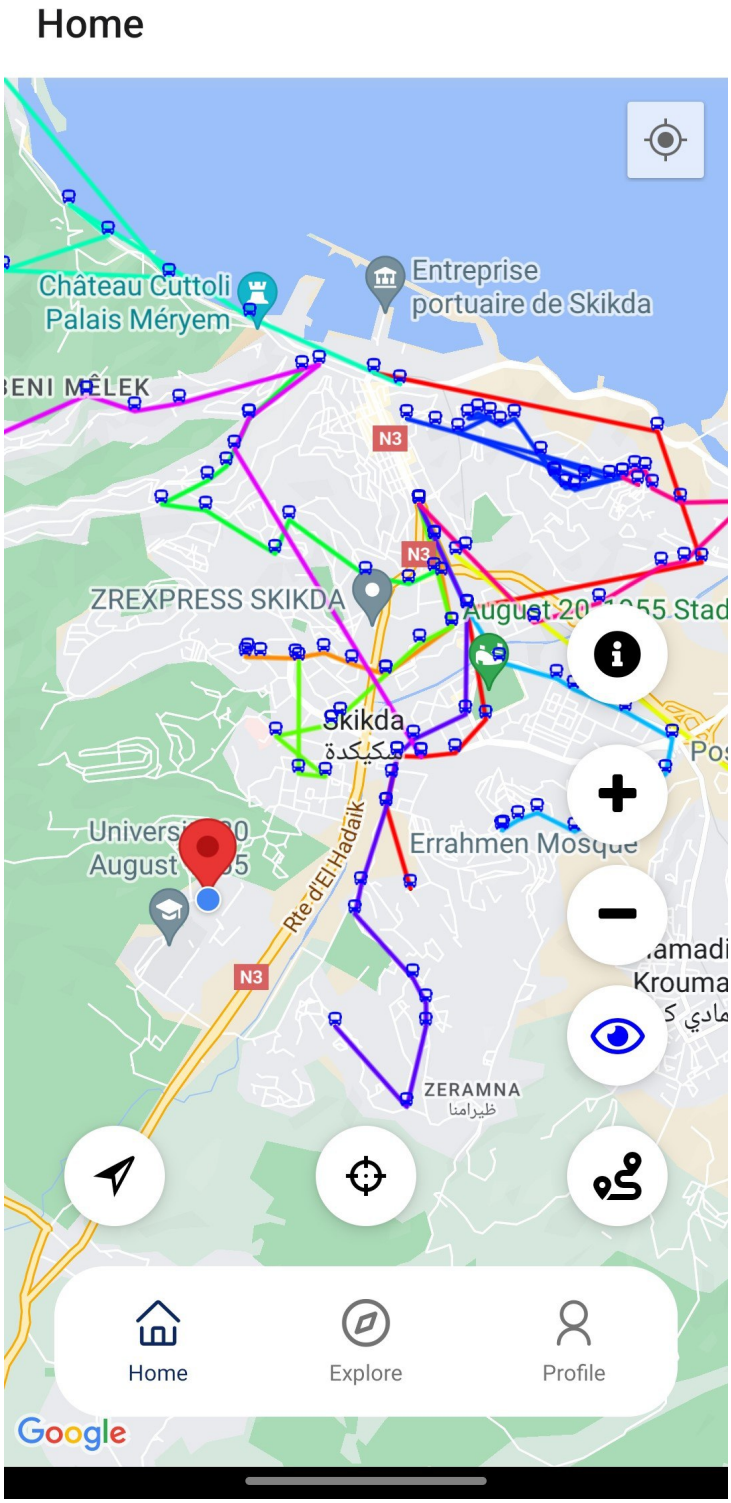
شكل 7.6: الصفحة الرئيسية

توضح هذه الصورة الصفحة الرئيسية لتطبيقنا. من هنا، يمكنك التنقل في الخريطة لتحديد موقعك الحالي ووجهتك. تحتوي الصفحة الرئيسية على دليل لاستخدام التطبيق، ومعلومات إضافية، وأزرار تكبير وتصغير، وخيارات لعرض وسائل النقل المختلفة في منطقتك، وزر التوجيه، واختيار الوجهة، وشريط تنقل للوصول بسهولة إلى الصفحة الرئيسية أو الملف الشخصي أو قسم الاستكشاف.



شكل 8.6: واجهة العثور على المسار الأقصر

توضح هذه الصورة أقصر مسار للحافلة بعد تحديد المواقع البداية والوجهة، وتعرض أيضًا محطات الحافلات على طول الطريق.



شكل 9.6: واجهة جميع وسائل النقل المتاحة

تعرض هذه الصورة جميع وسائل النقل المتاحة ومساراتها الخاصة.

المحور 7 نموذج العمل التجاري

شكل 1.7: نموذج العمل التجاري

خاتمة

في الختام، يُعد حل مشكلة النقل والتنقل من الأمور الملحة التي تتطلب تدخلًا فوريًا. ومع ذلك، واجهنا العديد من الصعوبات في معالجتها بسبب نقص الموارد والأبحاث، حيث أن هذه المشكلة جديدة ولا توجد لها سوابق بحثية كافية. قمنا بتجميع البيانات وتنظيمها يدويًا بسبب عدم توفرها بشكل مناسب، وواجهنا تحديات كبيرة تتعلق بكثرة البيانات وصعوبة معالجتها وجمعها بشكل جذري. يتطلب ذلك وقتًا طويلًا بالإضافة إلى نقص البيانات المفصلة في بعض المناطق، خاصة مع قلة الموارد والبحوث المتاحة.

من بين المشاكل التي واجهتنا، كان هناك افتقار للحلول المُجربة والنماذج المماثلة السابقة، مما يستلزم بذل جهود إضافية في البحث والتطوير. كما عانينا من نقص الدراسات والأبحاث المتخصصة في مجال النقل والتنقل الذكي، بالإضافة إلى نقص التمويل والإمكانيات المادية والبشرية اللازمة لمعالجة المشكلة بفعالية.

جمع البيانات من مصادر متعددة، مثل بيانات حركة المرور وأنماط السفر واحتياجات المستخدمين، وتنظيمها بطريقة تُسهل تحليلها واستخدامها، يتطلب وقتًا كبيرًا. لذا نسعى جاهدين للتعاون مع شركات النقل والحكومات لتبادل الخبرات وتطوير حلول متكاملة تساعد في حل مشاكل البيانات.

كما نواجه تحديات إضافية تتعلق بتحديث الخرائط وبناء مسارات جديدة تكون أقصر وأكثر كفاءة. على الرغم من هذه التحديات، تُبذل جهود حثيثة لمعالجتها من خلال جمع البيانات وتنظيمها والتعاون مع الجهات المختصة مثل شركات النقل والحكومات والمؤسسات.

ومع ذلك، فإن الالتزام الجاد بحل هذه المشكلة يبشر بإمكانية التوصل إلى حلول فعالة في المستقبل.