INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Isochrone Maps in Real Estate Geographic Information

JORGE MANUEL AGUIAR PESSOA
Setembro de 2024

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Isochrone Maps in Real Estate – Geographic Information

Jorge Pessoa

A dissertation submitted in partial fulfillment of
the requirements for the degree of Master of Science,
Specialization Area of Computer Systems

Supervisor (ISEP): Rosa Reis

Co-Supervisor (DEVSCOPE): David Mota

Porto, January 2024

Acknowledgments

Firstly, I would like to express my gratitude to DevScope for the opportunity to collaborate with such a great team, with a special thanks to my supervisor, David Mota, for his support and motivation throughout this project.

Furthermore, I would also like to express my gratitude to my academic supervisor Rosa Reis from ISEP for her insightful guidance, which was invaluable to the success of this work.

Finally, I am grateful to my family and friends for their unwavering encouragement and support along the way.

To those who have always supported and believed in me.

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

have not plagiarized or applied any form of undue use of information or falsification of

results along the process leading to its elaboration.

Therefore, the work presented in this document is original and authored by me, having not

previously been used for any other end.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P. PORTO.

ISEP, Porto, 15 de setembro de 2024

Abstract

The real estate industry is constantly evolving, yet traditional property recommendation systems often ignore crucial factors like accessibility and convenience, focusing predominantly on property features such as price, size, and location. To address this, this thesis explores the use of isochrone maps to enhance the personalization of real estate recommendations.

By examining the current shortcomings of recommendation systems, this research emphasizes how isochrone maps can offer a more complete understanding of a property's spatial accessibility. Through the creation of a prototype system, the study illustrates how incorporating isochrones into the recommendation process enables more informed property searches, assisting users in finding homes that not only align with their preferences but also suit their lifestyles regarding commute and convenience.

The results indicate that isochrone maps significantly improve real estate recommendation systems, providing a more user-friendly and thorough property search experience. This research covers the way for the real estate industry to offer more relevant, accessible, and satisfying recommendations to potential buyers.

Keywords: Real Estate Recommendations, Isochrone Maps, Spatial Accessibility

Resumo

A indústria imobiliária está em constante evolução, contudo, os sistemas tradicionais de recomendação imobiliária ignoram frequentemente fatores cruciais, como acessibilidade e conveniência, concentrando-se predominantemente em características como preço, dimensão e localização. Para enfrentar essa lacuna, esta tese explora o uso de mapas de isócronas como forma de melhorar a personalização das recomendações imobiliárias.

Ao examinar as limitações dos sistemas de recomendação atuais, esta pesquisa destaca como os mapas de isócronas podem proporcionar uma compreensão mais abrangente da acessibilidade espacial de uma propriedade. Através da criação de um sistema protótipo, o estudo demonstra como a incorporação de isócronas no processo de recomendação permite buscas por imóveis mais informadas, auxiliando os utilizadores a encontrar habitações que não só correspondam às suas preferências, mas também se adequem ao seu estilo de vida, em termos de deslocações e conveniência.

Os resultados indicam que os mapas de isócronas melhoram significativamente os sistemas de recomendação imobiliária, proporcionando uma experiência de pesquisa de imóveis mais amigável e completa. Esta investigação abre caminho para que o setor imobiliário ofereça recomendações mais relevantes, acessíveis e satisfatórias aos potenciais compradores.

Palavras-chave: Recomendações Imobiliárias, Mapas Isócronas, Acessibilidade Espacial

Contents

List of Figures

Supervisor (ISEP): Rosa Reis

Co-Supervisor (DEVSCOPE): David Mota

Porto, January 2024

List of Tables

Supervisor (ISEP): Rosa Reis

Co-Supervisor (DEVSCOPE): David Mota

Porto, January 2024

List of Code Snippets

Acronyms and Symbols

Acronyms

API	Application Programming Interface
FR	Functional Requirements
GIS	Geographical Information Systems
OSM	OpenStreetMap
ORS	OpenRouteService
RERS	Real Estate Recommendation Systems

1 Introduction

1.1 Background

The real estate industry is inherently dynamic and complex, constantly evolving to meet the demands of a growing population and increasing urbanization. In the current world where everyone is looking for a home that meets his or her needs and preferences, the need for efficient and effective real estate services has never been felt. Typically, property search and recommendation systems have relied on basic qualitative parameters including price, size, and area. However, these methods do not consider the aspect of accessibility and convenience which are critical factors in today's world (Droj et al., 2024).

The modern breakthroughs in the field of GIS and machine learning have started to change the existing state of affairs in the sphere of real estate recommendations. These technologies have improved the efficiency of the recommendation of properties by combining geographical information with data analytics. However, even today, most of the recommendation systems are based only on the intrinsic property features of the products and fail to consider the important factors such as accessibility and convenience to the potential buyers (Gharahighehi et al., 2021).

An interesting solution for these shortcomings is the application of isochrone maps. Isochrone maps are graphical illustrations of places that can be accessed within a given time from a particular place using different means of transport. These maps help the users to gain a good understanding of spatial accessibility and assess the distances of such facilities as service centers, transport terminals, and recreational facilities to potential properties (Curtis & Scheurer, 2016). Thus, using isochrone maps as a component of real estate recommendation systems can improve the decision-making for people who are in search of properties that would meet their lifestyle and utility requirements.

1.2 Problem Statement

Existing real estate recommendation systems are quite good in evaluating the basic criteria such as price, area, size, and location, but they cannot capture the essence of the constantly changing and highly subjective user preferences. Such systems generally work under a narrow set of parameters that focus on inherent property characteristics, while the indispensable aspects of access and proximity that would correspond to the buyer's schedule and needs are usually left unaddressed (Gharahighehi et al., 2021).

This gap between what buyers truly need and what current systems offer can lead to recommendations that, while consistent with the expressed needs of buyers, do not address the overall and more complex needs of potential buyers (Choi, 2023).

The contemporary dynamic world requires that aspects like the location of the buildings in relation to workplaces, schools, hospitals and other recreational facilities to be included in the recommendation systems of real estate. Sadly, many of the existing systems are unable to include these elements into their algorithms, which creates a vast gap between the offered properties and real needs and goals of potential (Droj et al., 2024).

Therefore, the incorporation of isochrone maps into real estate recommendation systems offers a good point to overcome these drawbacks. In the same way that isochrone maps help to transform how we think about spatial accessibility by presenting real-time travel-time information from any starting point. This is a shift from the traditional approaches of measuring the environment around a property and how it affects the lives of potential buyers and users (Curtis & Scheurer, 2016). Therefore, the integration of isochrone maps to bridge this gap in existing systems is important in improving the relevance and accuracy of recommendations that is required to assist the real estate industry to meet the current and future needs of the sophisticated and diverse market.

1.3 Objectives of the Study

The primary objective of this study is to investigate how isochrone maps can be integrated into real estate recommendation systems to improve the accuracy, convenience, and personalization of property recommendations. By doing so, the study aims to bridge the gap between traditional property recommendation methods, which focus on intrinsic property features, and the growing need for accessibility and convenience in real estate searches. To achieve this, the study sets out to accomplish the following specific objectives:

Investigation of Current Limitations:

- Perform a thorough examination of current real estate recommendation systems in order to recognize and comprehend the drawbacks resulting from their prevailing emphasis on intrinsic property attributes.
- Examine the implications of this restriction, especially with regard to undervaluing convenience and accessibility in fulfilling the changing expectations of prospective purchasers in terms of lifestyle.

Exploration of Isochrone Maps' Capabilities:

- Examine the complexities of isochrone maps and their potential to remedy the shortcomings of the existing recommendation systems.
- Analyse the ways in which isochrone maps can provide a more complex depiction of geographic accessibility by considering variables like closeness to transit hubs, recreational centre, and necessary services.

Prototype Development:

- Develop a real estate recommendation system prototype that incorporates isochrone maps into its algorithm.
- Ensure that the prototype system is not only technologically robust but also user-friendly.

Evaluation of Prototype Effectiveness:

- Conduct a rigorous evaluation of the prototype system's effectiveness in comparison to traditional real estate recommendation systems.
- Access the impact of isochrone map integration on the accuracy and personalization of property recommendations.

By pursuing these goals, this study hopes to provide insightful information to the real estate sector, providing a more advanced and comprehensive method of property suggestions that surpasses traditional criteria. The ultimate goal is to bridge the gap between user expectations and system capabilities, elevating the real estate recommendation process to a new standard of accuracy and personalization.

1.4 Research Questions

The research will be guided by the following questions:

Current Limitations in Real Estate Recommendations:

- What are the specific issues of current recommendation systems for real estates concerning individuality and relevance?
- How do these limitations affect the relevance of the recommendations provided to users?

Isochrone Maps' Potential to Address Limitations:

- How does the use of isochrone maps help to solve or even eliminate these limitations?
- What are the particular advantages of employing isochrone maps to improve the customization and effectiveness of the real estate recommendations?

Impact of Isochrone Maps on Recommendation Systems:

- To what degree does the addition of isochrone maps to a recommendation system affect its effectiveness and its users' satisfaction?
- In what way does the use of isochrone maps affect the general perception of users in property searches?

By addressing these research questions, this study intends to investigate the transformative potential of isochrone maps, shed light on the complexities of existing real estate recommendation systems, and offer a strong evaluation framework for contrasting the performance of systems with and without isochrone map integration.

1.5 Significance of the Study

The significance of this study extends beyond its immediate research focus, encompassing broader implications for the real estate industry, technological integration, and user experience. The key facets of the study's significance include:

- Revolutionizing Real Estate Recommendations: The implications of this study are that there will be a shift from how real estate recommendations are made illustrating the importance of isochrone maps in improving precision and individualization. Isochrone maps provide a new angle that is more in tune with the ever-changing and diverse nature of the demand of the property hunters today.

- Holistic Understanding of Property Surroundings: Isochrone maps in general offer a more holistic view of the property environment and offer a novel approach to the evaluation of accessibility. This outlook which includes location to social amenities like public transport, business places and recreation facilities helps potential customers to make wiser choices.
- Enabling Industry Adaptation: The conclusion of this study could encourage the improvements and flexibility among the organizations. Thus, the study provides a way forward for integrating the advanced spatial analysis forms into the case contexts of the real estate recommendation systems and showing how isochrone maps can help to overcome the problems of the existing systems.

- Enhancing User Experience and Satisfaction: Enhanced real estate recommendations due to isochrone maps are likely to be of great value in benefiting the end users greatly. Recommendations that are more precise and that take into account both stated needs and other often unvoiced needs and wants of the users will enhance satisfaction and the link between buyers and their dream properties.

- Guiding Future Research and Development: The findings of this study can help the future research of isochrone maps in the recommendations of the real estate sector and contribute to the further investigation of new technologies and approaches in the real estate market. The methods and findings of this study can be used as reference works for subsequent researches to enhance the use of spatial analysis to offer customized properties.

Conclusively, this research holds the prospect of highly impacting the real estate industry to show how isochrone maps can enrich real estate advice. Isochrone maps which represent the distance from the property that can be reached in a given time are a fresh approach to the accessibility and may present a more complete picture of the environment and thus the recommendations given will be more appropriate to the individual.

1.6 Organization of the Thesis

This section outlines the structural framework of the thesis. The content is organized to ensure clarity and coherence in presenting the research findings. The thesis unfolds in the following manner:

Chapter 1: Introduction: The introduction chapter provides the reader with the background and rationale for the study. This section provides the background to the study, research questions, objectives and significance of the study, and ethical concerns. It serves as a background to the subsequent chapters by pointing out the relevance of isochrone maps in real estate advice.

Chapter 2: Research Methodology: The research methodology chapter outlines the research design, method and approach that was adopted in the study. It involves the literature review, data collection techniques, system development and the evaluation. The chapter also outlines the criteria used in identifying information sources and the testing procedure that was used in assessing the efficiency of the system.

Chapter 3: State of the Art: The state-of-the-art chapter presents the literature review and technological background of real estate recommendation systems and isochrone maps. It also revisits the use of geographical information systems (GIS), isochrone maps, and APIs in enhancing the recommendation systems and also outlines the main drawbacks of the conventional systems.

Chapter 4: Analysis and Design: The analysis and design chapter deal with the system analysis and design, the domain model, use case diagram, and the functional and non-functional requirements. This paper outlines the isochrone-based real estate recommendation prototype and offers information on the system, actors, and interactions involved in the design and architecture of the system.

Chapter 5: Implementation: The implementation chapter describes the backend configuration with ORS and frontend development with React and Leaflet. The chapter takes the reader through the code that is used in rendering maps, capturing user input and creating isochrones.

Chapter 6: Solution Evaluation: The solution evaluation chapter compares the results of the isochrone-based recommendation system to answer the research questions. It evaluates the isochrone maps' accuracy, system scalability, usability, and the overall effect on real estate recommendations compared to the conventional systems.

Chapter 7: Conclusion: The last chapter provides a synthesis of the main findings, conclusions and the contribution of the research. It shows the degree to which the research questions were addressed, considers the findings' relevance to real estate recommendation systems, and outlines further research avenues.

1.7 Ethical Concerns

Regarding ethical concerns, these guide all phases of the project and are of paramount importance. The research process is in accordance with the Code of Good Practices and Conduct of P. Porto, specifically articles six, eight, and ten (Diário da República, 2020). Additionally, the software engineering process follows the ACM/IEEE-CS Software Engineering Code of Ethics (Gotterbarn et al., 1997), which outlines the responsibilities of software engineers to prioritize the public good, maintain high standards of professional competence, and ensure fairness and impartiality in the development of software systems.

In alignment with these ethical standards, the project addresses the following key areas:

- **Privacy Measures:** Although this system is in the prototype stage, privacy remains a top priority. Any data involved will comply with the company's privacy policies, ensuring secure and responsible handling.
- **Reliability:** The system is designed to deliver accurate and up-to-date information, using reliable data sources to ensure that the real estate recommendations it provides reflect real-world conditions.
- **Impartial Evaluations:** All evaluations conducted as part of this project will be carried out objectively, without bias or external influence.
- **Acknowledgment of Contributions:** All external tools and contributions will be properly credited, acknowledging the valuable work of the wider development community.

1.8 Project Timeline

The chronogram (Figure 1) provides a structured timeline, guiding the systematic progression of this research project and ensuring efficient management of resources for timely completion.

	2023 - 2024									
Task Description	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
Problem Identification and Objectives										
Literature review and identification of gaps										
System design										
Development of the prototype										
User testing and evaluation										
Analysis of results and final adjustments										

Figure 1 - Project Timeline

2 Research Methodology

This section details the rigorous research process, emphasizing academic integrity and adherence to ethical standards. A comprehensive literature review was conducted to identify existing knowledge and gaps, using strict inclusion and exclusion criteria.

2.1 Comprehensive Literature Review

The literature review is an essential component of any research project, as it provides a comprehensive and critical overview of the existing knowledge and gaps in a specific field of inquiry. As stated in the article "Literature reviews as independent studies: guidelines for academic practice" (2022) (Kraus et al., 2022), a literature review is a survey of scholarly sources on a specific topic. It provides an overview of current knowledge, allowing you to identify relevant theories, methods, and gaps in the existing research. This highlights how important it is for a literature evaluation to identify pertinent ideas, approaches, and areas of unfulfilled study needs.

Furthermore, the paper "Reviewing the research methods literature: principles and strategies illustrated by a systematic overview of sampling in qualitative research" (2016) (Gentles et al., 2016) notes that the literature review represents the most crucial step of the research process in qualitative, quantitative, and mixed research studies. This emphasizes how crucial a literature review is for setting the research in the context of other scholars and researchers and for illustrating how the current study fills a vacuum or advances a larger discussion.

2.2 Inclusion and Exclusion Criteria for Source Selection

Strict inclusion and exclusion criteria were used in the literature review to maintain high standards. Table 1 below illustrates how the systematic removal of unnecessary studies was achieved through an iterative process of exclusion based on particular criteria.

Table 1 - Research criteria

Criteria	Justification
Inclusion	Articles that mention real estate recommendation systems, GIS, and isochrone maps.
	Articles that reflect practical cases of using GIS and isochrone maps in real estate.
	Articles that present different methods and techniques for implementing isochrone maps in real estate recommendation systems.
	Articles that mention user preferences and personalization in real estate recommendations.
Exclusion	Commercial publications.
	Articles not related to real estate recommendation systems, GIS, and isochrone maps.
	Articles that do not show evidence of the author's perspective or methodology.

2.3 Information Sources

In the process of identifying pertinent literature on the given topic, an extensive search was conducted across several scholarly databases, including MDPI (Multidisciplinary Digital Publishing Institute), Google Scholar, and IEEE Xplore. Relevant information and examples on GIS and isochrone maps were also sought from reputable websites such as Geoapify and Geospatial World.

2.3.1 Research Terms

The papers incorporated in this thesis for its state-of-the-art research were gathered using the following terms:

- Real estate recommendation systems
- Isochrone maps in GIS
- Spatial accessibility in real estate
- Machine learning in real estate systems
- Geographic Information Systems (GIS) for real estate

By utilizing these search terms, a comprehensive collection of relevant studies was obtained to support the analysis of the current state of real estate recommendation systems.

2.3.2 Selection Process

A thorough methodology was used during the systematic review and selection process to ensure the inclusion of the most relevant studies. The selection procedure is outlined in the following steps:

- **Initial Gathering:** The project's scope was initially covered in an in-depth collection of papers and articles gathering a number of 11,960 results.
- **Exclusion based on Criteria:** A review was conducted to eliminate articles that did not meet the requirements.
- **Final Selection:** After the previous steps, a final list of 45 results is presented in accordance with the criteria.
- **Final Utilization:** Of the final list, 15 articles were selected (Table 2).

Table 2 – Articles selected

Title	**Author(s)**	**Year**
GeoAI: Integration of Artificial Intelligence, Machine Learning, and Deep Learning with GIS	Choi, Y	2023
The Use of Machine Learning in Real Estate Research	Choy, L. H. T., & Ho, W. K. O.	2023
Mapping evacuation risk on transportation networks using a spatial optimization model. Transportation Research Part C: Emerging Technologies	Church, R. L., & Cova, T. J.	2000
PLANNING FOR PUBLIC TRANSPORT ACCESSIBILITY: An International Sourcebook. In Planning for Public Transport Accessibility: An International Sourcebook	Curtis, C., & Scheurer, J.	2016
A Comprehensive Overview Regarding the Impact of GIS on Property Valuation	Droj, G., Kwartnik-Pruc, A., & Droj, L.	2024
Destination recommendation systems: Behavioural foundations and applications.	Fesenmaier, D. R., Wöber, K. W., & Werthner, H.	2006
Accessibility evaluation of land-use and transport strategies: Review and research directions.	Geurs, K. T., & van Wee, B.	2004
Recommender Systems in the Real Estate Market—A Survey	Gharahighehi, A., Pliakos, K., & Vens, C.	2021

OpenStreetMap: User-Generated Street Maps. Pervasive Computing	Haklay, M., & Weber, P.	2008
Geographic Information: Science, Systems, and Society	Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W.	2015
Real-time routing with OpenStreetMap data. GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems	Luxen, D., & Vetter, C.	2011
Equity of urban service delivery: A comparison of different accessibility measures.	Neutens, T., Schwanen, T., Witlox, F., & de Maeyer, P.	2010
Measuring accessibility: Positive and normative implementations of various accessibility indicators.	Páez, A., Scott, D. M., & Morency, C.	2012
Potential Path Areas and Activity Spaces in Application: A Review.	Patterson, Z., & Farber, S.	2015
Isochrone-Based Accessibility Analysis of Pre-Hospital Emergency Medical Facilities: A Case Study of Central Districts of Beijing	Zhao, Y. ;, Zhou, Y., Mu, L., Yang, J., Kainz, W., Zhao, Y., & Zhou, Y.	2024

3 State of Art

This section provides a detailed exploration of isochrone maps, their various applications, and the technologies that support their functionality, particularly in real estate recommendation systems.

3.1 Introduction to Isochrone Maps

Isochrone maps, as highlighted by (Curtis & Scheurer, 2016), are a valuable tool in geographic and urban analysis, providing a visual representation of areas that can be reached from a specific point within a designated time frame using various modes of transportation (Figure 2). These maps create boundaries where all points within are accessible within a set travel time from a central location, regardless of distance. This practical measurement of accessibility, as noted by (Geurs & van Wee, 2004), addresses real-world travel conditions, making them more functional than traditional distance-based maps.

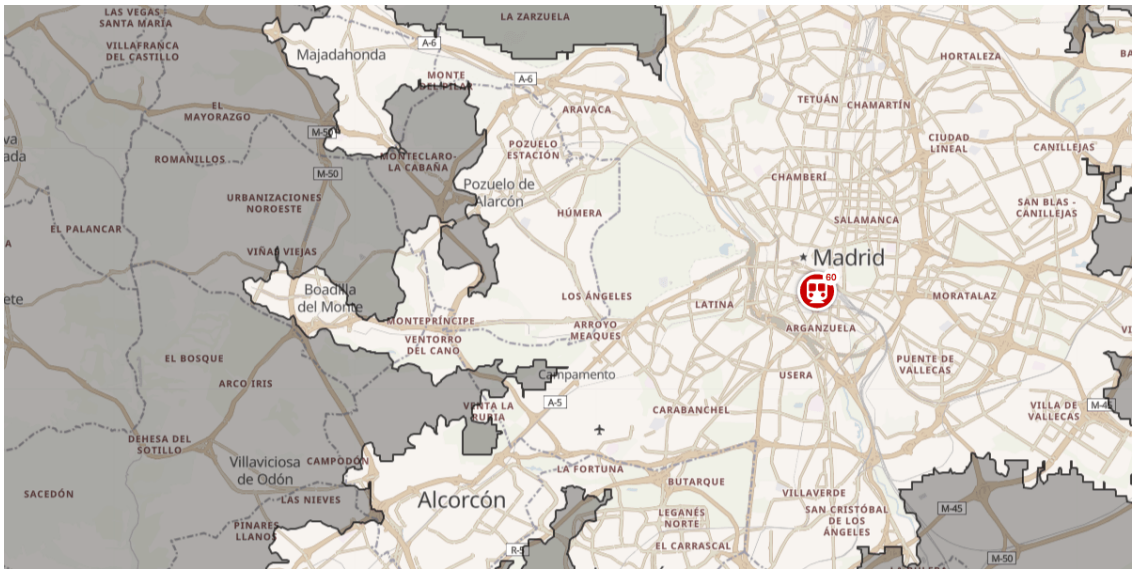

Figure 2 - Isochrone Map (Isochrone Maps: Get Real Reachability Times - Geoapify, 2024)

Recent advancements in Geographic Information Systems (GIS) have dramatically improved the production/usage of isochrone maps. The complex datasets (like road networks, traffic conditions and topography) that the isochrones generated by using GIS technology includes make their accuracy better than any other method. Isochrones were in fact a commonly used mapping technique within urban planning and public service

delivery historically. As (Curtis & Scheurer, 2016) explain, city Planners use these maps when assessing how well residents in various regions can access essential services, such as hospitals, schools or transport hubs to make sure infrastructure meets the needs of its population.

In the realm of real estate, isochrone maps have recently gained popularity as a tool to visualize property accessibility based on potential buyers' daily commutes or proximity to key amenities. Unlike simple distance-based searches, isochrone maps, as described by (Páez et al., 2012), consider actual travel times, making them a more reliable indicator of a property's accessibility. The generation of isochrone maps typically relies on routing algorithms that calculate the shortest or fastest routes between points, considering various factors such as traffic, road types, and transportation modes.

3.2 Current Applications of Isochrone Maps

Isochrone maps have evolved beyond their traditional uses in urban planning and have found applications in a variety of fields. Their ability to visually represent accessibility within a given time frame makes them invaluable for decision-making in areas such as public transportation, emergency services, environmental analysis, and real estate (Figure 3). This section explores some of the most significant current applications of isochrone maps, highlighting their versatility and the innovative ways in which they are being utilized.

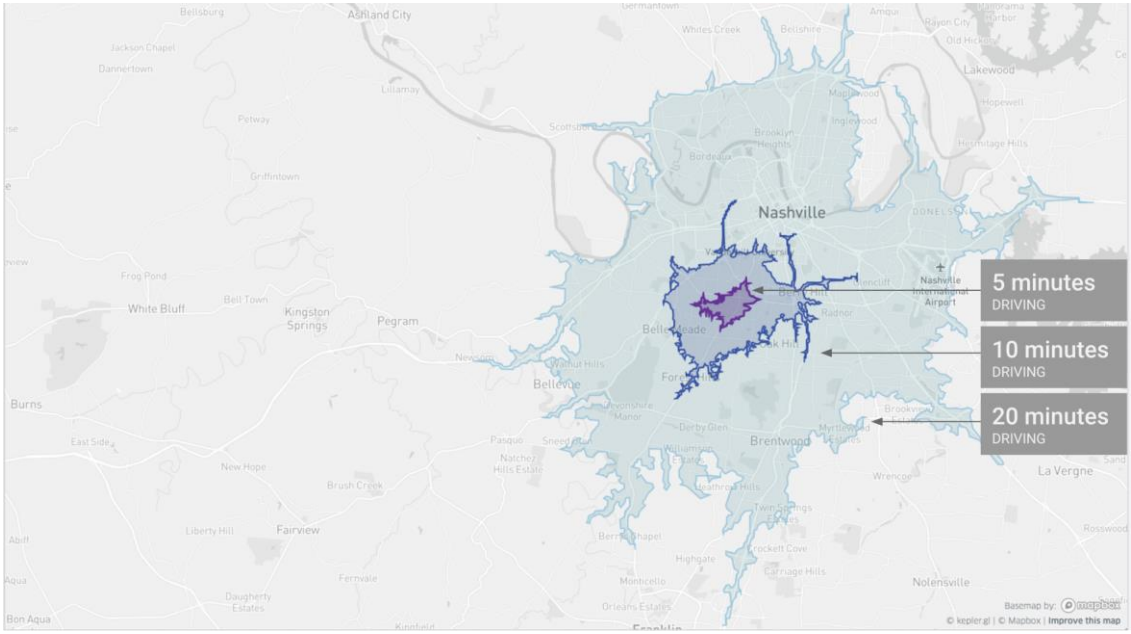

Figure 3 - Isochrone Map – Area vs Time (Mapbox, 2023)

3.2.1 Urban and Transportation Planning

Isochrone maps are frequently used in urban and transportation planning to determine and optimize the access to different public services. (Curtis & Scheurer, 2016) emphasize their role in ensuring that critical services (i.e., hospitals, schools and public transportation) are accessible in different neighborhoods throughout the city. (Geurs & van Wee, 2004) argue that accessibility evaluations using isochrone maps are critical in optimizing public transport routes and reducing overall commute times in urban areas. These maps allow transportation organizations to ensure that residents can access key locations within the intended time frame.

3.2.2 Emergency Response and Disaster Management

Time is often the most critical factor, in emergency response and disaster management. (Church & Cova, 2000) explain that isochrone maps can be used to estimate time of response for emergency services. By illustrating regions that can be arrived in by a certain critical response time period, those with the power to make decisions will have better coverage & efficiency results. A city might use isochrone analysis to determine the best locations for new fire stations, making sure that all neighborhoods are within reach in a timely enough manner. Furthermore, (Zhao et al., 2024) illustrate how isochrone maps can be used in the context of pre-hospital emergency medical facilities. Their study on the central districts of Beijing demonstrates how isochrone-based accessibility analysis can improve emergency response by identifying optimal locations for facilities and defining efficient evacuation paths. Additionally, isochrone maps can contribute to disaster management by supporting in the definition of paths for evacuation and identification of more susceptible regions according their distance from safety or shelter zones.

3.2.3 Environmental and Sustainability

Environmental researchers and urban ecologists have been using isochrone maps to analyze how accessible green nodes, such as parks or nature reserves within a city are. (Neutens et al., 2010) discuss how urban planners use isochrone maps to increase environmental friendliness by enhancing accessibility to green spaces, thus improving residents' quality of life. They are: In studies that look at the effects of pollution or climate change on different sites to see how far adverse impacts might spread over time, under specific circumstances; in those which critique and analyses migration patterns across an urban area as well. For instance, researchers can apply isochrone maps to simulate the propagation of contamination across a stream network through time and thereby contribute towards more effective environmental policies.

3.2.4 Real Estate and Retail Location

Isochrone maps are now a critical component of the user interface for both buyers and sellers across real estate. As (Páez et al., 2012) note, these maps help potential buyers see where houses are located in relation to places like work, school or retail centers. By not just measuring distance but time taken this is a truer reflection of property values as it factors in real world travelling times, for example traffic. Isochrone maps are used by real estate developers and retailers to pinpoint sites that have good accessibility for their client base. A business might use isochrone research to decide where the best place for a store would be if surrounded by their target demographic within 15 minutes' drive time.

3.2.5 Tourism and Recreation

Companies that are involved in the tourism regularly apply Isochrone mapping to further enhance their customer experience. These maps are used by travel firms and tourism boards to show how long a trip between different attractions can take from the center, thereby enabling tourists to better plan their holiday activities, as (Fesenmaier et al., 2006) highlights. In this case, a map of isochrones could be used showing the different activities that are within a 30-minute car drive from their hotel business and then attract tourists who want to minimize travel time during their duration. Outdoor recreation businesses could also use isochrone maps to highlight the proximity of hiking trails, lakes or other natural features so consumers can better plan for day trips.

3.3 Technological Background

The development and application of isochrone maps are deeply intertwined with advancements in geographic information systems (GIS), computational geometry, and routing algorithms. This section delves into the technological foundations that make the creation and utilization of isochrone maps possible, highlighting the key tools, methodologies, and platforms that have emerged as standards in the field.

3.3.1 Geographic Information Systems (GIS)

The advancement of isochrone mapping has relied heavily on the innovations made in Geographic Information Systems (GIS). With Geographic Information Systems (GIS) platforms, it's possible to store spatial data and represent a three-dimensional view of it allowing time-based accessibility graphs to be overlaid on standard maps. (Longley et al., 2015) explain that these advancements, including the ability to integrate real-time traffic data, make isochrones more representative of real-world conditions. These platforms have out-of-the-box algorithms for generating isochrones, enabling users to build accurate accessibility maps by setting various parameters such as travel mode and time.

3.3.2 Computational Geometry and Algorithms

The computation of isochrone maps can be considered as a computational geometry problem, the area of mathematics dealing with spaces and figures. As (Cormen et al., 2022) describe, traditional shortest path problems over networks, such as road systems (e.g., when you want the "shortest" route from A to B not just by distance but also by time), are then used for deriving isochrone estimates. The determination of these shortest pathways is predominately done using algorithms such as Dijkstra and A* (A-star). However, this isochrone calculation becomes a lot trickier for several reasons when you add things like traffic patterns or different forms of public transportation into the mix. This resulted in the creation of more advanced heuristics and algorithms that make isochrones to be generated even faster with a better quality. For instance, it cuts down computation time enormously by pre-processing the road network when creating isochrones, effectively eliminating unnecessary calculations through a technique called contraction hierarchies.

3.3.3 Routing Engines and APIs

Routing engines such as OpenRouteService, Google Maps API, and Mapbox provide the necessary infrastructure for computing isochrones based on real-world data. (Luxen & Vetter, 2011) explain that different routing algorithms and engines can significantly impact the accuracy and efficiency of routing tasks, which indirectly affects the quality of isochrone maps. Even though the Google Maps API is considered a gold standard for mapping services, its fees can become prohibitive at scale. While it offers robust geographic support, it lacks comprehensive real-time traffic information.

3.3.4 Data Sources and Quality

Isochrone maps are only as accurate as the data underneath them. This means mass transit schedules, road networks and traffic flow — all of which can have very divergent sources with coverage and quality that are spotty at best. OpenStreetMap (OSM) is highly utilized data source for routing and isochrone creation due to its global coverage, free access format. However, OSM is crowdsourced so the quality of data can vary especially in rural or less densely populated areas. (Haklay & Weber, 2008) highlight that while OSM provides a valuable resource for mapping, its variable quality necessitates caution. In order to mitigate this, many routing engines append their own commercial or proprietary datasets over OSM data for higher precision and detailed results. Applications requiring very fine accuracy require high-quality, real-time data sources (e.g., emergency response planning or detailed urban studies). Including satellite imaging and commercial traffic services or maybe some government transportation databases.

3.3.5 Computational Infrastructure

Large amounts of computer infrastructure are required to compute needed isochrone maps, especially in the case where large datasets exist or real-time applications are considered. To serve the massive processing required for isochrone generation, cloud computing platforms such as Microsoft Azure, Google Cloud and Amazon Web Services (AWS) provide scalable solutions. These layers allow to process real-time data stream processing (for e.g. vehicle tracking), to run complex algorithms on huge dataset at scale and provide a distributed way of generating isochrone maps for any part in the world, whenever needed around the globe. Moreover, enhancements in parallel processing and GPU computing have substantially shortened the process time to build isochrone maps enabling them as near real-time output for on-demand (ride-sharing, delivery services, etc.) applications (Armbrust et al., 2010).

3.4 Isochrone-based Real Estate Apps

The integration of isochrone maps into real estate platforms represents a significant advancement in how potential buyers and renters explore housing options. By combining geographic accessibility with property data, these applications offer a unique approach to real estate, focusing on travel times rather than just location. This section examines the current state of isochrone-based real estate apps, discussing the most notable examples, their features, and the technological innovations driving them.

3.4.1 Evolution of Real Estate Search Technologies

Most traditional real estate search tools revolved around location-based searches and allowed users to view homes by refining their selection based on features such as price, size, or closeness to amenities. Borrowing heavily from the work of accessibility academics, some method conglomerations addressed portions of this missing time-space dimension for modern urban residents who appreciate travel times as well as access on daily basis but these have tended to fall short in one aspect or another. This changed with isochrone-based searches in real estate apps that allowed users to find homes based on trip times from a certain area, making house hunting more user centric, as highlighted by (Choy & Ho, 2023). These innovations utilize machine learning to improve property recommendations by considering travel times and user preferences (Gharahighehi et al., 2021).

Early implementations of isochrone-based searches were limited by computational capabilities and the availability of accurate routing data. For a long time, the constraints were too significant, but thanks to recent advances in routing engines and access to high quality real-time traffic data this is no longer true. Modern apps now offer dynamic isochrones that adjust in real-time based on current traffic conditions, further enhancing the relevance and accuracy of search results, as noted by (Zhao et al., 2024).

3.4.2 Isochrone Maps and Real Estate Recommendation Systems

A large number of current real estate recommendation systems are based primarily on static property characteristics, including price, area, or the number of bedrooms, without taking into account such factors as accessibility and convenience. Such systems are based on machine learning and data analysis, as mentioned by (Choy & Ho, 2023), but do not take into account such essential parameters as the time it takes to get to work and availability of services.

Furthermore, while GIS has been used in urban planning and accessibility of transport, the use of GIS in real estate is still limited. According to (Droj et al., 2024), GIS is becoming increasingly important in generating spatial analysis across different fields; however, real estate systems are still fairly immature in terms of their capacity to incorporate these dynamic layers of GIS. The main issue with existing recommendation systems is that they do not consider travel time and distance to the key amenities while recommending properties.

3.4.3 Comparative Evaluation of Existing Approaches

Current real estate recommendation systems are mostly distance-based and are centered on distance to particular points of interest (e.g., schools, hospitals). Such systems may employ the Euclidean distance or simple driving distances to estimate the attractiveness of a location. However, this approach doesn't consider important factors like traffic congestion, current traffic situation, or multimodal transport options. According to (Gharahighehi et al., 2021) the current systems have started to take into consideration the user preferences, however the criteria used are static and do not change depending on the conditions that occur in daily accessibility.

Isochrone maps present a time-based approach, which contrasts with distance-based methods by measuring accessibility in terms of travel time, not distance. This approach offers a more realistic understanding of accessibility by considering real-world variables like traffic and public transport schedules. Studies like those conducted by (Páez et al., 2012) emphasize that time-sensitive accessibility indicators are far more relevant to decision-making, as users are more likely to prioritize how long it takes to reach important locations rather than how far they are.

However, the application of isochrone maps in real estate is not well developed. As demonstrated by (Luxen & Vetter, 2011), real-time routing algorithms can be used in developing dynamic isochrones but there is still a significant void in the integration of these technologies in real estate platforms. Almost all the real estate platforms either employ the static isochrones that do not change according to the existing conditions such as traffic or do not employ them at all. In addition, systems that do utilize isochrone maps tend to confine their use to a specific mode of transport (for example, driving) while not

providing access to multiple modes of transport which are very important within urban environments where walking, cycling, and the use of public transport are all important means of commuting.

3.4.3.1 Key Limitations in Existing Systems

Despite some progress, most real estate recommendation systems still suffer from several key limitations:

1. Narrow Focus on Intrinsic Property Features: The majority of systems prioritize property characteristics such as price and size over contextual factors like accessibility. As (Choy & Ho, 2023) state, these systems miss out on the holistic view of what makes a property valuable—its connection to the surrounding environment.

2. Lack of Real-Time Data Integration: Most of the systems fail to use real-time data to give the latest accessibility information.

3. Multimodal Transport Ignorance: Most systems focus exclusively on car travel, neglecting the fact that many people in cities use public transport, bike or walk. As highlighted by (Curtis & Scheurer, 2016), understanding multimodal travel options is essential for effective urban planning, and this principle can extend into real estate recommendations.

3.4.3.2 Potential of Isochrone Maps to Address Limitations

The use of isochrone maps is one of the major opportunities which can help to overcome these limitations. These are more dynamic and user-friendly property recommendation maps that include real time data on traffic conditions, available public transport schedules, and other forms of transport. (Zhao et al., 2024) have demonstrated the effectiveness of isochrone maps in emergency medical planning, showing that time-based accessibility is crucial. By applying this principle to real estate, isochrone-based systems can deliver more relevant recommendations, particularly for buyers who place a premium on commuting times and convenience.

3.4.4 Technological Innovations in Isochrone-based Real Estate Apps

Isochrone based real estate platforms have been made possible thanks to several key technological innovations. To begin with routing engines that the platform could integrate (OpenRouteService, Google Maps API) these platforms now offer very precise isochronous maps which will be auto adjust. Through a complex set of algorithms, they determine commute times that take into account numerous variables — from breakdowns in traffic to how frequently buses run down our street network, leaving people with timely and exact information (Luxen & Vetter, 2011).

Secondly, their delivery as a cloud-based services made existing systems scalable and responsive to the above requirements. Thanks to the cloud infrastructure, it is possible for real estate apps to process all of those calculations that generate an Isochrone map in real time (Armbrust et al., 2010).

Lastly, advancements in user interface (UI) and user experience (UX) design have made isochrone-based searches more engaging and easier to use, contributing to their growing popularity.

3.4.4.1 Limitations and Challenges

While isochrone-based real estate apps have a lot of advantages, using them also comes with some challenges and limitations. A major limitation is the requirement for current, accurate data. Although routing engines like OpenRouteService and Google Maps API offer good data, occasionally they may make mistakes due to outdated maps or unforeseen traffic. However, generating real-time isochrones imposes a computational load which could lead to performance issues especially in high-resolution road networks or heavy traffic areas (Haklay & Weber, 2008).

An additional issue is the merger between several types of transportation to a single isochrone map. Many already provide travel times for vehicle and public transportation but adding walking, cycling, or others to the same isochrone can become messy underneath and less accurate. Future developments in multimodal routing algorithms may help address this issue (Patterson & Farber, 2015).

3.5 Routing Engines for Isochrone Maps

Routing engines are the backbone of isochrone map creation, determining the optimal paths between points based on various parameters such as distance, time, and transportation mode. Several routing engines are commonly used in the industry, each offering distinct features, advantages, and limitations. The choice of routing engine can significantly influence the accuracy and functionality of the generated isochrones.

Popular Routing Engines

1. **Google Maps API** (Google Maps Platform Documentation | Routes API | Google for Developers, 2024)
    - **Pros**: Highly accurate, extensive global coverage, real-time traffic data, ease of integration.
    - **Cons**: Commercial product with usage fees, limited customization, reliance on proprietary data.

- **Use Case**: Ideal for commercial applications where accuracy and real-time data are critical, such as in logistics and ride-sharing services.

2. **Mapbox** (Mapbox | Maps, Navigation, Search, and Data, 2024)
    - **Pros**: Customizable, high-quality maps, strong developer support, offers both free and paid tiers.
    - **Cons**: Limited historical traffic data, can be expensive at scale.
    - **Use Case**: Best for applications requiring a high degree of map customization and integration with other Mapbox tools.
3. **OpenRouteService** (Openrouteservice, 2024)
    - **Pros**: Open-source, cost-effective, highly customizable, supports various modes of transportation, including walking, cycling, and driving.
    - **Cons**: Less accurate in some regions compared to commercial alternatives, limited real-time traffic data.
    - **Use Case**: Ideal for research projects, non-commercial applications, and when customization or specific route calculations are needed.
4. **GraphHopper** (GraphHopper Directions API with Route Optimization, 2024)
    - **Pros**: Fast route calculation, open-source, supports a variety of transportation modes, customizable.
    - **Cons**: May require more setup and customization effort, limited commercial support.
    - **Use Case**: Suitable for applications requiring fast processing and flexibility in routing algorithms.

OpenRouteService is selected for this project as it is open-source, and therefore there is a lot of flexibility in the ways that the API can be adapted to suit the needs of the application. They embrace multiple transportation modes which are rather important concerning the actual construction of isochrone maps elaborated in accordance with various transportation scenarios such as driving, cycling, walking, etc. Furthermore, OpenRouteService is based on OpenStreetMap, and its navigation data is highly detailed and flexible because it perfectly fits complex research and numerous applications that need to define specific routing settings. While it does not have up-to-date traffic data as

some existing commercial products such as Google Maps, it is affordable and can be easily customized to be the best solution to develop a user-oriented real estate application.

3.6 Review of APIs for Implementing Isochrone Maps in Real Estate Applications

The implementation of isochrone maps in real estate applications has been facilitated by various APIs. This section provides an in-depth evaluation of two widely used APIs – Mapbox GL and Leaflet – with a focus on their features, advantages, and how well they meet the specific requirements of real estate applications.

3.6.1 Mapbox GL

Mapbox GL stands as a robust Software Development Kit (SDK) for web maps, hailing from the innovative domain of Mapbox. Renowned for pioneering vector maps technologies, Mapbox GL naturally supports the Mapbox Style Specification for vector maps. This library not only opens doors to more opportunities but also offers a rich array of data visualization options, expanding the horizons of vector maps. However, a noteworthy development is the shift to a non-free license in February 2021, introducing new considerations for users. While providing extensive opportunities and supporting vector maps, Mapbox GL may offer less direct control compared to Leaflet's solution (Mapbox | Maps, Navigation, Search, and Data, 2024).

Pros:

- Expansive opportunities for data visualization.
- Native support for Mapbox Style Specification for vector maps.

Cons:

- Shifted to a non-free license in February 2021.
- May offer less direct control compared to Leaflet's solution.

3.6.2 Leaflet

Leaflet emerges as an open-source JavaScript API designed for interactive maps. Boasting a simple interface, excellent documentation, and a vibrant community, Leaflet facilitates rapid results and efficient problem-solving. However, it is crucial to note that Leaflet is no longer actively developed. Despite this, its straightforward interface,

coupled with robust community support, makes it an attractive option for certain real estate applications (Leaflet - a JavaScript Library for Interactive Maps, 2024).

Pros:

- Simple interface and great documentation.
- Large community support.

Cons:

- No longer actively developed.
- Core functionality lacks native support for vector tiles.

3.6.3 Comparison of Mapbox GL and Leaflet in Isochrone Map Integration

When comparing Mapbox GL and Leaflet, it is important to note that both MapboxGL and Leaflet have their strengths and weaknesses. The choice between the two depends on the specific requirements of the real estate application. For applications that require detailed and interactive isochrone maps, MapboxGL might be the better choice due to its support for vector maps. On the other hand, for applications that prioritize speed and simplicity, Leaflet might be more suitable.

3.7 App's Unique Approach

In the evolving landscape of real estate search technologies, the app introduces a unique approach by placing a high emphasis on multi-modal mobility options. Unlike traditional real estate platforms that often limit users to searching based on static location-based criteria, the app enables users to explore housing options based on dynamic isochrones that consider various modes of transportation, including walking, cycling, and driving.

3.7.1 Multi-modal Mobility Focus

One of the notable features of this app is that it can generate isochrones based on such means of transport as car, bike, and public transport. Additional, users can enter the mode of transport they intend to use and the application will suggest properties that are within a certain radius of the selected transport means. This approach not only does not take into account the transport needs of a user but also encourages user to make right lifelong sustainable choices where one can walk or cycle if possible.

For example, a user wanting a house within a 15-minute cycling distance from his workplace can easily do so. The app will then make a map with a pin pointing to the current location and all other properties within the said radius brought to light. It becomes particularly useful in the urban environments, where cycling or walking can be faster and much friendlier in terms of the environment, to driving. By including such modes, the app provides a versatile solution to meet the needs of modern home seekers who often need transportation to get to the location.

3.7.2 Time-sensitive Property Suggestions

Another key feature of the app is its ability to suggest properties based on precise travel times rather than just geographical distance. This is very useful for persons who consider commuting or their proximity to basic facilities. Instead of the exact travel time, the app adapts from the general time to demonstrate a more realistic view of the accessibility that goes a long way in determining housing.

For example, the app can pinpoint homes that are 20 minutes' drive or 10 minutes' walking distance from a particular point. Such information helps to provide clients with proper property recommendations, corresponded to their lifestyle and time management preference, and, thus, contribute to users' more satisfactory decisions making.

3.7.3 Summary

In sum, the app distinguishes itself by proposing the approach based on multi-modal mobility and property recommendations within the given time range. Incorporating the various forms of transport, which includes walking, cycling and driving makes the app more person oriented and environmentally friendly when compared to seeking for property. Further, it eliminates the trivial differences in the geographical distances and offers better decisions based on the time required for commute and choice of the user. Thus, this kind of approach adds to the quality of a user experience and also provides a more effective and a sustainable way of searching for properties.

4 Analysis and Design

This section presents a general description of the system architecture and the main design decisions made during the development process to meet the requirements of the isochrone-based real estate recommendation system. It covers the key components, actors, requirements, and technology selections that drive the system's development.

4.1 Domain Model

The domain model serves as a blueprint for the system, defining the key entities, their attributes, and relationships. It is crucial for understanding the structure of the data and how different components of the system interact with one another. In this real estate recommendation system, the domain model (Figure 4) highlights the main entities such as Users, Properties, Owners, and Locations, and their interconnections.

User: This entity collects the data of the users of the system including their name, email, gender, nationality, preferences, and search history. It means that the User can search for several properties, and it is implemented in the model by the one-to-many relationship between the User and Property entities.

Property: This entity holds some characteristics of the real estate listings including the price, type, size, number of bedrooms and bathrooms, description and images. Each property is linked to a Location and an Owner.

Owner: This entity is to capture the owners of the properties that are listed in the system, it could be individuals or other organizations. The Owner has details such as name, email, and phone number and is related to many properties, as shown by the one-to-many relationship between the Owner and Property.

Location: This entity stores information about geographical location of each property such as city, address, state, latitude and longitude. between them is one to one because each property in the system has its own location.

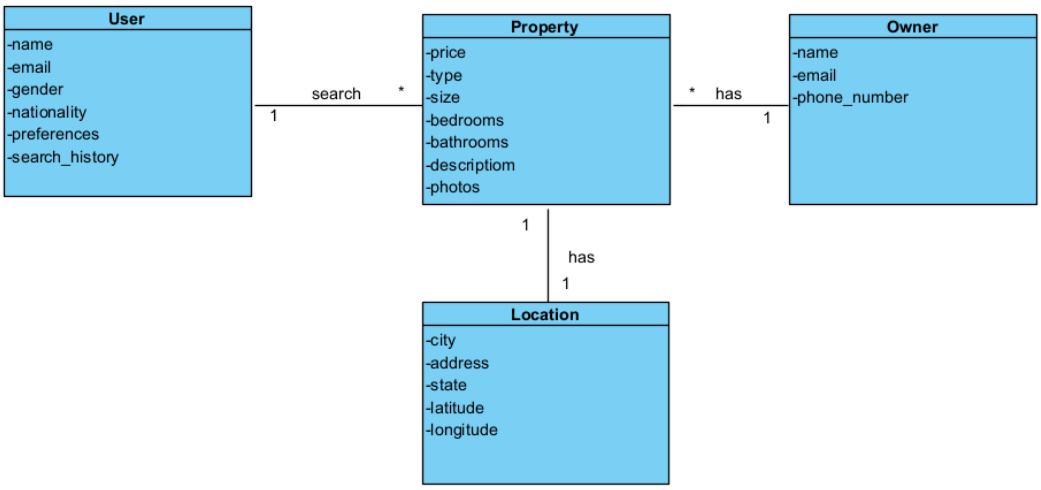

Figure 4 – Isochrone-based Real Estate Recommendation System Domain Model

4.2 System Actors

In system design, actors refer to the different players or the roles that are involved in the interaction with the system whether as a person, organization or other systems. These actors have a significant importance in specifying the functional requirements of the system as they represent the users and other external entities interacting with the system. It is crucial to define and analyze these actors to guarantee that the system meets all the required use cases and interaction efficiently (Baszuro & Swacha, 2020) .

The importance of having well-defined system actors is it guarantees that all the interactions are accounted for and the system is designed with the users in mind. If these actors are identified and defined at the early stages of the system design, the designers will be in a position to understand what the users require, how the system will interface with other systems and how to overcome any challenges that may be encountered in the integration process (Baszuro & Swacha, 2020).

Here are the actors defined for the system:

User

The primary user who searches for properties based on preferences such as location, price, and property features. This actor interacts with the system to find suitable properties and assess their accessibility through isochrone maps.

Property Owner

A person or agency responsible for listing properties on the platform. This actor manages property information and ensures that the property details (price, photos, description, etc.) are up to date.

Administrator

Manages the overall platform and ensures system reliability and data accuracy. This actor oversees user accounts, property listings, and platform performance.

External Systems (APIs, GIS Services)

External services that provide additional functionality, such as generating isochrone maps or retrieving traffic and location data. These systems interact with the real estate platform to provide real-time geographic information.

4.3 Requirements Engineering

Requirements Engineering is a process that takes place in the development of software where the necessary requirements for a system to meet its objectives are identified, documented and managed (Hoy & Xu, 2023). This process is very important as it provides a way in which the stakeholders (users, developers and clients) have a common ground on what the system should do and how it should do it.

Functional requirements describe the specific functionalities that the system must provide. On the other hand, non-functional requirements define the system's operational characteristics, such as performance, security, and scalability.

4.3.1 Functional Requirements

FR1: The system shall generate isochrone maps based on the user’s selected location, travel mode, and time range.

FR2: The system shall integrate the property search with the isochrone maps, allowing users to filter and view properties that fall within the generated isochrone area.

4.3.2 Non-Functional Requirements

Scalability: The system must be capable of scaling to handle an increasing number of users and search queries, particularly when deployed on cloud infrastructure.

Performance: Isochrone generation and property search results should be provided in real-time or near real-time, ensuring a responsive user experience.

Security: User data must be securely stored and transmitted, with appropriate authentication and authorization mechanisms in place.

Usability: The system should have an intuitive user interface, allowing users to easily navigate and utilize the application.

Availability: The system should ensure high availability, particularly during peak usage times, through the use of load balancing and failover strategies.

Maintainability: The system should be designed for easy maintenance, allowing for updates and bug fixes without significant downtime.

4.4 Design Alternatives and Technology Considerations

To ensure that the system design meets the requirements of flexibility, scalability, and usability, various alternatives were evaluated. These alternatives are grouped into three key categories: Isochrone Map Generation Tools, Frontend Frameworks, and Mapping Libraries. Each category was assessed based on criteria such as cost, customizability, scalability, accuracy, and ease of integration.

4.4.1 Isochrone Map Generation

This section considers tools that generate isochrone maps, a core feature of the recommendation system.

Google Maps API

- **Advantages**: Offers real-time traffic data, highly accurate, global coverage.
- **Disadvantages**: Expensive for high usage, limited customization for isochrone generation.

OpenRouteService (ORS)

- **Advantages**: Open-source, customizable, supports multiple transportation modes, based on OpenStreetMap data.
- **Disadvantages**: No real-time traffic data.

Mapbox API

- **Advantages**: High-quality maps, supports vector maps, customizable.
- **Disadvantages**: Costs can rise significantly with usage.

4.4.2 Frontend Frameworks

Different frontend frameworks were considered for building the user interface of the real estate recommendation system.

React

- **Advantages:** Strong ecosystem, widely adopted, high performance, and excellent integration with mapping libraries like Leaflet.
- **Disadvantages:** Steeper learning curve compared to other frameworks, especially with TypeScript.

Vue.js

- **Advantages**: Simpler learning curve, lightweight, and fast.
- **Disadvantages**: Smaller ecosystem compared to React.

Angular

- **Advantages**: Fully-featured framework, offers built-in tools for testing, routing, and state management.
- **Disadvantages**: Larger and heavier framework, higher complexity, longer development time.

4.4.3 Mapping Frameworks

Mapping frameworks were evaluated for their ability to render **isochrone maps** and integrate with the chosen frontend technology.

Leaflet

- **Advantages**: Open-source, lightweight, easy to integrate with React and Vue.js, good plugin ecosystem, great for custom maps.
- **Disadvantages**: Limited support for 3D maps or advanced visualizations.

Mapbox GL

- **Advantages**: High-quality vector maps, good support for mobile and web, real-time data visualization options.
- **Disadvantages**: Cost increases with usage, complex to set up.

Google Maps

- **Advantages**: Extensive global coverage, highly accurate, provides real-time traffic data and comprehensive API support.
- **Disadvantages**: High costs for API usage, low flexibility in custom map generation.

4.4.4 Evaluation Grid

Table 3 - Comparative Analysis of Isochrone Map Generation Tools, Frontend Frameworks, and Mapping Libraries

Criteria	Cost	Customizability	Scalability	Ease of Use	Accuracy
Google Maps API	High	Low	High	High	High
ORS	Low	High	High	Medium	High
Mapbox API	High	Medium	High	Medium	High
React	Low	High	High	High	N/A
Vue.js	Low	High	Medium	High	N/A
Angular	Medium	Medium	High	Medium	N/A
Leaflet	Low	High	High	High	N/A
MapBoxGL	Medium	Medium	High	Medium	N/A
Google Maps	Medium	Medium	High	High	N/A

4.4.5 Selection Justification

After evaluating the various alternatives in terms of cost, customizability, scalability, ease of use and accuracy, the following key technologies were chosen for the design and implementation of the system:

OpenRouteService (ORS): ORS was chosen for the generation of isochrone as it is open source which made it flexible and cheaper to use. In contrast to Google Maps API, for example, ORS provides multi-modal route options (e. g., walking, cycling, driving) and is fully adaptable to the requirements of the system in question.

React: React was selected as the frontend framework due to having a strong community support, the component-based approach, and possibility to scale. Furthermore, the

integration of React with the Leaflet library made the process of rendering maps and managing users' interactions more effective and easier.

Leaflet: The mapping library used in the development of this application was chosen to be Leaflet due to is light weight, simplicity and ability to integrate with both React and ORS, compared to Mapbox or Google Maps, which are more functional and expensive, as well as requiring more time and effort to implement, Leaflet offers a simple solution for displaying interactive 2D maps and working with isochrone data. This is an open-source map that is easy to integrate and since the project required mapping it was the best option without having to pay for it.

4.5 System Architecture

4.5.1 Use Case Diagram

The use case diagram (Figure 5) provides a visual representation of the interaction between the user and the system, focusing on the primary functionality: searching for properties using isochrone maps.

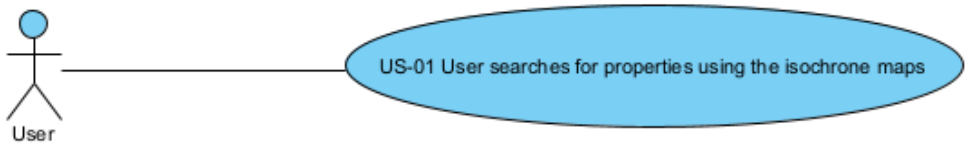

Figure 5 - Use Case Diagram

The use case "User searches for properties using the isochrone maps" captures the main interaction between the user and the system. The user enters their search criteria, such as location, and selects the desired travel mode and time range. The system then generates an isochrone map based on these inputs, highlighting the areas that can be reached within the specified time frame. Properties located within the isochrone are then displayed to the user.

A scenario example could be:

A user wants to find properties within a 30-minute drive from their workplace.

1. The user enters the workplace location into the system, selects "Driving" as the travel mode, and sets the travel time frame to 30 minutes.

2. The system generates an isochrone map showing all areas that can be reached within 30 minutes driving.
3. The user views the properties within this isochrone area.

4.5.2 Sequence Diagram

The sequence diagram provides a dynamic view of the system, illustrating the sequence of interactions between different components as a user performs a specific action—in this case, searching for properties using isochrone maps. This diagram helps to visualize the order of operations and how data flows through the system, ensuring that all components work together to achieve the desired functionality (Koç et al., 2021).

Use Case 01: User Searches for Properties Using Isochrone Maps

The main goal of this use case is to enable the user to search properties within a given travel distance from a chosen location. The sequence diagram (Figure 7) shows how the frontend application communicates with the OpenRouteService (ORS) to create isochrone maps and present properties.

Sequence of Interactions:

1. User Interaction: The sequence starts with the User engaging the Frontend Application through the map and setting the travel parameters like mode of transport (for example, walking, cycling, driving) and time (for instance, within 15 minutes).

2. Frontend Processing: The Frontend Application receives the inputs from the user and makes a request to the OpenRouteService (ORS). The request contains the coordinates of the selected location, the mode of transport, and time period.

3. Request to OpenRouteService (ORS): When the Frontend Application receives the request it processes it to conform to the ORS API before sending an API request to the OpenRouteService. This request is made to ORS to produce an isochrone map that has been determined by the specifications.

4. Isochrone Generation by ORS: The OpenRouteService (ORS) then processes the request and produces an isochrone map that shows the area that can be reached within the stipulated time from the users chosen location. It should then be noted that ORS returns this isochrone data to the Frontend Application.

5. Frontend Renders Isochrone: After getting the isochrone data from the ORS, the Frontend Application analyzes the data and then display the isochrone on the map

using the Leaflet. The isochrone is depicted on the map as a shaded area which gives the impression of the accessible area.

6. Property Search and Display: At the same time, the Frontend Application refines the list of available properties to those, which are located in the isochrone area created. These filtered properties are then displayed to the user on the map to enable the user to search as well as select properties that are accessible to them.

7. User Reviews Results: the User reviews the displayed properties, which is now overlaid with the isochrone map. The user is able to select a property to view more information including the description and other features of the property.

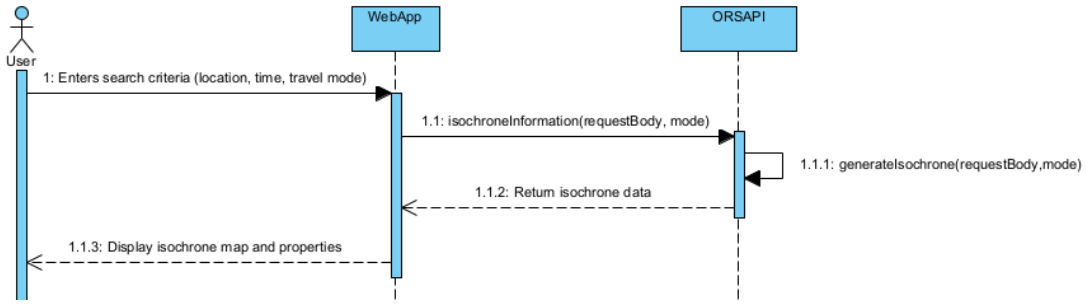

Figure 6 - Sequence Diagram

4.5.3 Logical View

The Logical View of a system offers a clear picture of how a certain system is supposed to work in terms of its components. In this project, the Logical View is focused on the user and the frontend application that interacts with the OpenRouteService (ORS) instance (Bass et al., 2003).

Since the architecture of the frontend is quite straightforward and it is directly connected to ORS, the Logical View highlights the path of data and the functions of different parts in the system. The frontend application is designed to receive inputs from the user including the location and the mode of travel and display the results on a map. The OpenRouteService works as the backend, which processes these inputs and returns the isochrone maps which are then displayed on the frontend.

This simple architecture makes the system very effective and easy to manage since all the computational work is done by ORS while the frontend is used to facilitate the user's interaction. The interactions highlighted in this Logical View will take place at the container level to show how the system components support the real-time delivery of property accessibility information to the user (Figure 8).

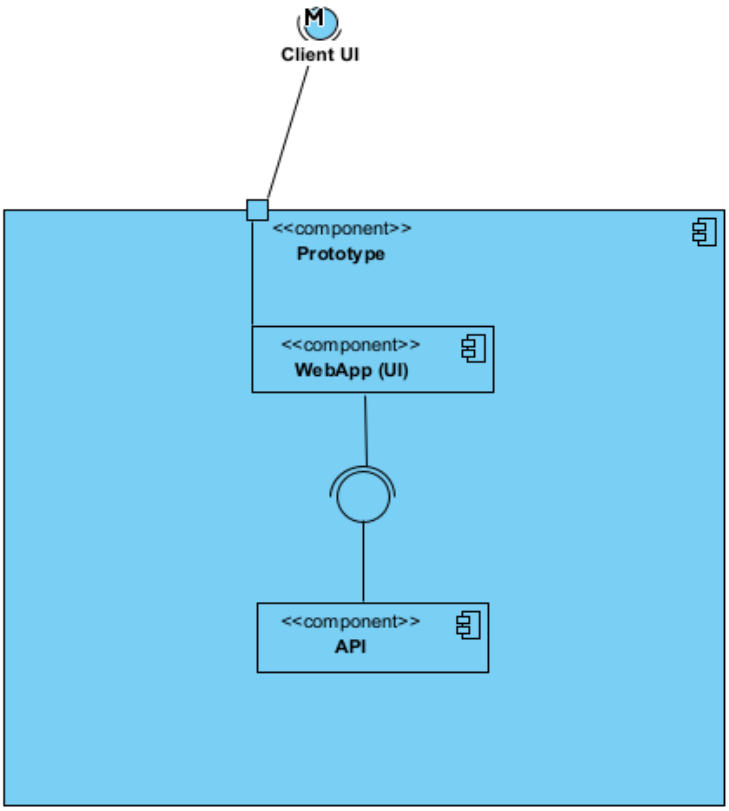

Figure 7 - Logical View

This view ensures that the system's architecture is both efficient and easy to maintain, with ORS handling all computational tasks and the frontend focusing on user interaction. The interactions in this are presented at the container level, illustrating how the components work together to deliver real-time property accessibility information to the user.

5 Implementation

This section presents the overview of the implementation process and presents the important components and configurations that were used for the purpose of achieving the functionality of the Isochrone Map-based real estate recommendation system. The section starts with the description of the main components of the OpenRouteService (ORS), which is an open-source routing engine that forms the basis of the application's geospatial analysis capabilities. Given the flexible and robust nature of ORS, the implementation required primarily adapting and customizing its configuration to meet the specific requirements of this project, rather than developing new algorithms from scratch.

The implementation process is further described by explaining the changes made to the ORS configuration files, to further refine the routing profiles and fine-tune performance to the identified geographical area of interest. Further, the section focuses on the integration of ORS with the frontend application built in React that communicates with the routing engine to produce and visualize isochrone maps.

The user stories presented in Section 4.3.3 were addressed, demonstrating how each functional requirement was implemented within the system. This involves the configuration of the Docker environment that contains the ORS, this makes the deployment process to be consistent across different environments. The frontend application is then discussed, which shows how it was developed to integrate with the ORS API in order to let the user create and visualize isochrones for property recommendation based on travel time.

Finally, this section provides a brief description of the testing strategies used to ensure the correctness of the system. These tests were intended to check the functionality of the integration between the frontend and the backend services so that the end-users can get the right results at the right time. With the help of an open-source routing engine, the implementation process was less time-consuming and the main effort was made on how the system should be configured and fine-tuned to provide the user with a more personalized and contextual experience.

5.1 Backend Implementation: OpenRouteService (ORS) Setup

The backend of the application is powered by ORS, which was chosen due to its robust routing capabilities and open-source nature. The primary objective was to configure ORS to support the generation of isochrone maps based on user-defined parameters such as travel time and mode of transport. This involved several key steps:

5.1.1 Data Collection with OpenStreetMap

OpenStreetMap (OSM) was used as the primary data source for routing information. Specifically, the portugal.osm.pbf file was downloaded from OpenStreetMap, which contains detailed geographic data for Portugal. This data is crucial for the ORS to generate accurate routes and isochrones. The OSM data is highly detailed and community-driven, making it a reliable and up-to-date source for geographic information.

```
engine:
  source_file: /home/ors/files/portugal.osm.pbf
  init_threads: 1
  preparation_mode: false
  graphs_root_path: ./graphs
  graphs_data_access: RAM_STORE
  elevation:
    preprocessed: false
    data_access: MMAP
    cache_clear: false
    provider: multi
    cache_path: ./elevation_cache
```

Code Snippet 1 - OSM region configuration

5.1.2 Docker Configuration

The deployment of ORS was managed using Docker, which provided a consistent and isolated environment for the application. The Docker Compose file defined the necessary services, ensuring that the ORS API was exposed on the appropriate ports for interaction with the frontend application. The configuration allowed for flexibility, enabling the ORS to be easily adapted to different environments without manual reconfiguration.

```yaml
version: '3.8'

services:

  ors-app:
    build:
      context: ./
    container_name: ors-app
    ports:
      - "8080:8082"  # ORS API exposed on port 8080
      - "9001:9001"  # Additional monitoring port

    image: local/openrouteservice:latest

    volumes:
      - ./ors-docker:/home/ors  # Mount ORS application directory
for logs, graphs, etc.

    environment:
      REBUILD_GRAPHS: False  # Prevent unnecessary graph rebuilding
      CONTAINER_LOG_LEVEL: INFO  # Log level set to INFO
      XMS: 1g  # Initial Java heap size
      XMX: 6g  # Maximum Java heap size
```

Code Snippet 2 - Docker compose file

This setup facilitated the seamless deployment of ORS, ensuring that it could efficiently handle routing requests with the allocated system resources.

5.1.3 ORS Configuration

The ors-config.yml file is central to ORS's operation, dictating how routing data is processed and how isochrone maps are generated. Key parameters were configured to optimize the service for the geographical region of interest. The ORS configuration was tailored to support multiple transportation profiles, including driving, cycling, and walking. Each profile was adjusted to ensure optimal performance, including consideration of elevation and specific path restrictions.

Driving Profile

```
profiles:
  car:
    enabled: true
    profile: driving-car
    elevation: true
    encoder_options:
      turn_costs: true
      block_fords: false
      use_acceleration: true
    preparation:
      min_network_size: 200
      methods:
        ch:
          enabled: true
          threads: 1
          weightings: fastest
        lm:
          enabled: false
          threads: 1
          weightings: fastest,shortest
          landmarks: 16
        core:
          enabled: true
          threads: 1
          weightings: fastest,shortest
          landmarks: 64
          lmsets: highways;allow_all
    execution:
      methods:
        lm:
          active_landmarks: 6
        core:
          active_landmarks: 6
    ext_storages:
      WayCategory:
      HeavyVehicle:
      WaySurfaceType:
      RoadAccessRestrictions:
        use_for_warnings: true
```

Code Snippet 3 - Driving Profile

Cycling Profile

```
bike-regular:
  enabled: true
  profile: cycling-regular
  encoder_options:
    consider_elevation: true
    turn_costs: true
    block_fords: false
  elevation: true
  ext_storages:
    WayCategory:
    WaySurfaceType:
    HillIndex:
    TrailDifficulty:
```

Code Snippet 4 – Cycling Profile

Walking Profile:

```
walking:
  enabled: true
  profile: foot-walking
  encoder_options:
    block_fords: false
  elevation: true
  ext_storages:
    WayCategory:
    WaySurfaceType:
    HillIndex:
    TrailDifficulty:
```

Code Snippet 5 - Walking Profile

These parameters allowed the creation of isochrone maps that accurately represent travel times across various terrains and road conditions.

5.2 Frontend Implementation: React and Leaflet with TypeScript

The frontend was built using React and TypeScript to provide a type-safe and scalable environment for developing user interfaces. Leaflet, a widely used JavaScript library for interactive maps, was integrated through the react-leaflet package. This section explains how Leaflet was used to render maps, handle user interactions, and display isochrone data.

5.2.1 Map Rendering and Isochrone Calculation

Map Rendering

Leaflet was chosen for its simplicity and performance in rendering interactive maps. The react-leaflet library was used to integrate Leaflet with React, allowing for declarative map rendering and state management within the React component tree. Leaflet's extensive plugin ecosystem and ease of use made it an ideal choice for this application.

```
import 'leaflet/dist/leaflet.css';
<MapContainer center={center} zoom={13} style={{ height: '100%',
width: '100%' }} zoomControl={false}>
    <TileLayer
        url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
        attribution='© <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors'
    />
    <ZoomControl position="topright" />
    <Markers />
    {isochroneData && <Polygon positions={isochroneData} />}
    {randomPoints.map((point, index) => (
        <Marker key={index} position={point} icon={myIcon}>
            <Popup>
                <p style={{ fontSize: '14px', fontWeight: 'bold'
}}>{descriptions[index]}</p>
                <img src={images[index]} alt={`House ${index}`}
style={{ width: '250px', height: 'auto' }} />
            </Popup>
        </Marker>
    ))}
</MapContainer>
```

Code Snippet 6 - Map rendering with Leaflet

Components Breakdown:

- **MapContainer**: This sets the view of the map to the default center coordinates with a zoom level of 13. Also, the map uses OpenStreetMap **tiles** (TileLayer) for the background and provides interactive zoom controls
- **Isochrone Overlay:** If isochroneData is available a Polygon is drawn on the map showing the area accessible within the selected time frame.
- **Random Markers:** To simulate property locations, random markers (<Marker>) are placed and shown in the isochrone area. In each marker there is a description and image of the place, which popup after clicking it.

Isochrone Calculation

The calculateIsochrone function is responsible for communicating with the ORS API to fetch the isochrone data. It takes the user's selected location, time, and transportation mode as input and processes the response to display the isochrone.

```
const calculateIsochrone = async (id: string, position: number[],
time: number, mode: string) => {
    setIsLoading(true);
    const requestBody: IsochroneRequest = {
        id: id,
        locations: [[position[1], position[0]]],
        location_type: "start",
        range: [time * 60],
        range_type: "time",
        units: "m",
        options: {
            avoid_borders: "controlled"
        },
        area_units: "m",
        intersections: false,
        attributes: ["area"],
        interval: time * 60,
        smoothing: 0
    };
    try {
        const response = await
MapService.isochroneInformation(requestBody, mode);
        let coordinates =
response.features[0].geometry.coordinates[0];
        coordinates = coordinates.map((c: any[]) => [c[1], c[0]]);

        const polygon = turf.polygon([coordinates]); // Create a
polygon from the coordinates
        const randomPoints =
generateRandomPointsInsideIsochrone(polygon, 15); // Generate 15
random points


        setIsochroneData(coordinates);
        setRandomPoints(randomPoints);
    } catch (error) {
        console.error(error);
    }
    setIsLoading(false);
};
```

Code Snippet 7 - Isochrone calculation

The calculateIsochrone function sends a request to the ORS API, passing the user's current location, mode, and time frame. The API returns a polygon representing the area accessible within the specified time. Once the API response is received, the

isochroneData is updated, and the polygon is rendered on the map to visualize the isochrone.

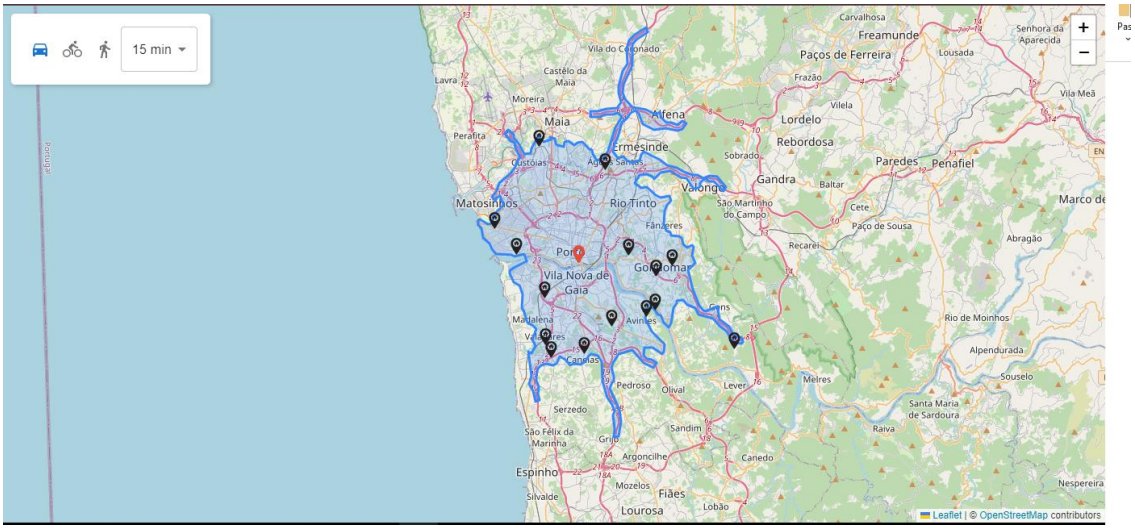

Figure 8 - Map Display with Isochrone Generated for a 15-Minute Drive

The map (Figure 9) shows the isochrone (highlighted area) representing the region that can be reached within 15 minutes by car from the selected location. The isochrone dynamically updates based on user inputs for travel mode and time frame.

5.2.2 Event Handlers for User Input

User inputs such as travel time and transportation mode are handled through specific event handlers. These event handlers trigger the recalculation of the isochrone based on the updated inputs, ensuring that users can dynamically adjust the map view.

```
const handleTimeChange = (event: any) => {
    const newTime = event.target.value;
    setTime(newTime);
    calculateIsochrone(markerPosition, newTime, mode);
};

const handleModeChange = (newMode: string) => {
    setMode(newMode);
    calculateIsochrone(markerPosition, time, newMode);
};
```

Code Snippet 8 - Event handlers

Event Handlers:

- **handleTimeChange**: This function is triggered when the user selects a new time frame (5, 15, or 30 minutes). It updates the time state and recalculates the isochrone using the calculateIsochrone function.
- **handleModeChange**: This function updates the selected mode of transportation (e.g., driving, cycling, or walking) when the user selects a new mode. It also recalculates the isochrone based on the updated travel mode.

These handlers allow users to dynamically adjust the map's isochrone based on their input, ensuring an interactive and responsive experience.

5.2.3 State Management

State management is crucial for ensuring that the user interface dynamically updates based on user input, system events, and API responses. The following state variables control the behavior of the isochrone map, travel mode, and time frame.

```
const IsochroneMap: React.FC<MapProps> = ({ center }) => {
    const [markerPosition, setMarkerPosition] = useState(center);
    const [isochroneData, setIsochroneData] = useState(null); //
State to store the isochrone data
    const [time, setTime] = React.useState(15);
    const [mode, setMode] = React.useState('driving-car');
    const [isLoading, setIsLoading] = React.useState(false);
    const [randomPoints, setRandomPoints] =
useState<LatLngTuple[]>([]);
```

Code Snippet 9 - State variables

State Variables:

- **markerPosition**: Tracks the user's selected position on the map. This position serves as the starting point for the isochrone calculation.
- **isochroneData**: Stores the coordinates of the isochrone polygon returned by the ORS API. This data is used to render the isochrone on the map.
- **time**: Represents the user-selected travel time (in minutes) for generating the isochrone. The default value is set to 15 minutes.
- **mode**: Tracks the selected transportation mode (e.g., driving, cycling). The default mode is driving ('driving-car').
- **isLoading**: A boolean flag used to indicate when the system is fetching or calculating the isochrone. While loading, user interactions are disabled to prevent conflicting requests.

- **randomPoints**: An array of randomly generated points (representing properties) that fall within the isochrone area.

By effectively managing the application state, the frontend is able to provide a dynamic and interactive experience, ensuring that changes in user input (such as time or mode) are immediately reflected in the isochrone data and map rendering.

5.2.4 MapService for ORS API Communication

The MapService is responsible for interacting with the ORS API. It sends the isochrone request and processes the response to return relevant data to the frontend. This service encapsulates the API interaction logic, ensuring clean and modular code within the frontend components.

```
import {BACKEND_ENDPOINT} from "../App";
import {IsochroneRequest} from "../types/Isochrone";

class MapService {
    async isochroneInformation(body: IsochroneRequest, profile:
string): Promise<any> {
        try {
            const response = await fetch(BACKEND_ENDPOINT +
'/v2/isochrones/' + profile, {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify(body),
            });
            // Check if the request was successful (status code 2xx)
            if (response.ok) {
                return await response.json(); // You can modify this
based on your needs
            } else {
                // If the request was not successful, handle the
error
                const errorData = await response.json();
                throw new Error(errorData.message);
            }

        } catch (error) {
            throw error;
        }
    }
}

export default new MapService();
```

Code Snippet 10 - Map Service

In the case of an error (network failure, incorrect parameters, or server-side issues), the catch block logs the error for debugging purposes and throws a user-friendly message. This ensures that the application doesn't crash, and users are provided with meaningful feedback if something goes wrong.

5.2.5 Type Definitions for Isochrone Request

The TypeScript interface defines the structure of the isochrone request, ensuring type safety and preventing errors during development. It provides clarity regarding what data is expected in the API request, and using strong types ensures better error handling and code maintenance.

```
export interface IsochroneRequest {
    id: string;
    locations: LatLng[];
    location_type: string;
    range: number[];
    range_type: string;
    units: string;
    options: RouteOptions;
    area_units: string;
    intersections: boolean;
    attributes: string[];
    interval: number;
    smoothing: number;
}

interface RouteOptions {
    avoid_borders: string;
}


type LatLng = [number, number];
```

Code Snippet 11 - Type Definitions for Isochrone Request

Explanation of Interfaces:

- **IsochroneRequest**: This interface defines the structure for the request sent to the ORS API. It includes:
    - **locations**: An array of coordinates (latitude and longitude) defining the starting point for the isochrone calculation.
    - **range**: Specifies the travel time (in seconds) used to calculate the isochrone.

- **RouteOptions**: This sub-interface defines additional options for the routing, such as whether borders should be avoided in the isochrone calculation.
- **LatLng**: A tuple that represents the latitude and longitude of the location.

5.2.6 Travel Mode and Time Frame Menu

The travel mode and time frame menu is a key part of the frontend interface, allowing users to select their preferred mode of transportation and travel duration.

The menu is implemented using a combination of Box, IconButton, and Select components from the Material-UI library. These components provide a clean, responsive UI for selecting travel preferences. Below is an explanation of the key parts of the code:

```
<Box sx={{ display: 'flex', alignItems: 'center' }}>
    <IconButton onClick={() => handleModeChange('driving-car')}
color={mode === 'driving-car' ? 'primary' : 'default'}
disabled={isLoading}>
        <DirectionsCar />
    </IconButton>
    <IconButton onClick={() => handleModeChange('cycling-regular')}
color={mode === 'cycling-regular' ? 'primary' : 'default'}
disabled={isLoading}>
        <DirectionsBike />
    </IconButton>
    <IconButton onClick={() => handleModeChange('foot-walking')}
color={mode === 'foot-walking' ? 'primary' : 'default'}
disabled={isLoading}>
        <DirectionsWalk />
    </IconButton>

    <Select
        value={time}
        onChange={handleTimeChange}
        displayEmpty
        inputProps={{ 'aria-label': 'Without label' }}
        disabled={isLoading}
    >
        <MenuItem value={5}>5 min</MenuItem>
        <MenuItem value={15}>15 min</MenuItem>
        <MenuItem value={30}>30 min</MenuItem>
    </Select>
</Box>
```

Code Snippet 12 - Travel Mode and Time Frame Menu

Components Breakdown:

Box: It's a container with flexbox property to make the child components (buttons and dropdown) to be in a horizontal position. The alignItems: 'center' property ensures that the buttons and the dropdown are aligned vertically in the middle of the box.

IconButton: There are three IconButton: for driving, cycling, and walking. The handleModeChange function is triggered when a user clicks a button passing the corresponding mode.

Disabled State: The buttons are disabled when the system is loading to avoid any interaction of the user during the processing of the background task.

Select and MenuItem: The dropdown enables the user to select the time interval (for example, 5, 15 or 30 minutes). The value property is set to the time state and the handleTimeChange function is called when a user interacts with the select element. The dropdown is also disabled when the value of the isLoading flag is true.

Functional Overview

Mode Selection: When the user clicks on any travel mode icon the handleModeChange function triggers and changes the current mode. This leads to a new generation of the isochrone based on the new mode of travel.

Time Frame Selection: Select component allows the user to select a time range which is then passed to the handleTimeChange function. This brings the state and changes the isochrone map depending on the selected travel time.

This code ensures that the interface is intuitive and responsive, updating the map based on user preferences while managing system state.

5.2.7 Travel Mode and Time Frame Selection UI

The interface presents a simple, user-friendly menu where users can select their preferred travel mode (e.g., driving, cycling, or walking) and specify the time frame they desire (e.g., 5, 15, or 30 minutes). The design focuses on ease of use, ensuring that users can easily adjust their preferences and visualize the isochrone map based on their choices.

The following figure (figure 9) displays the interface used for selecting the travel mode and time frame.

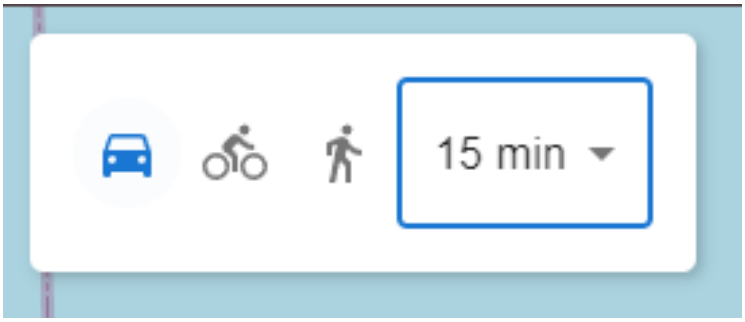

Figure 9 - Travel Mode and Time Frame select UI

The UI consists of:

- **Travel Mode Selection Icons**: Users can choose between various travel modes such as driving, cycling, and walking by selecting the corresponding icons.
- **Dropdown Menu for Time Frame Selection**: A dropdown menu that allows users to define the time range, offering options such as 5, 15, or 30 minutes.

5.2.8 Isochrone Maps Comparison for Different Travel Modes

This section demonstrates how the travel modes impact accessibility by comparing the isochrones generated for driving, cycling, and walking. The time taken for each mode is assumed to be 15 minutes but the area covered varies significantly due to the differences in speed and route availability.

Driving Mode.

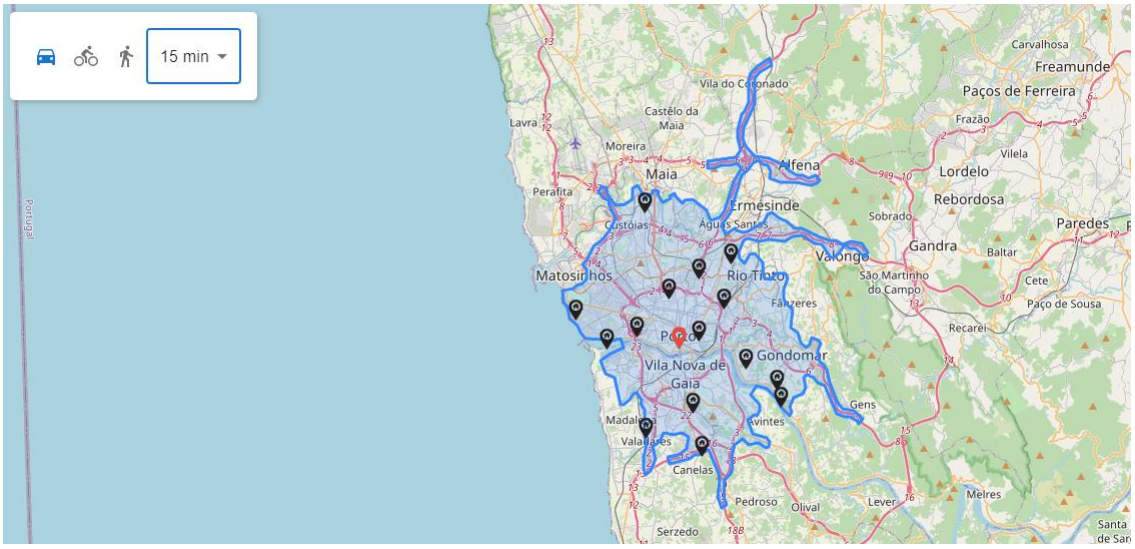

Figure 10 – Isochrone generated by driving with a time of 15 minutes

This mode typically covers the largest area, reflecting the higher speed and network of roads available for cars.

Cycling Mode

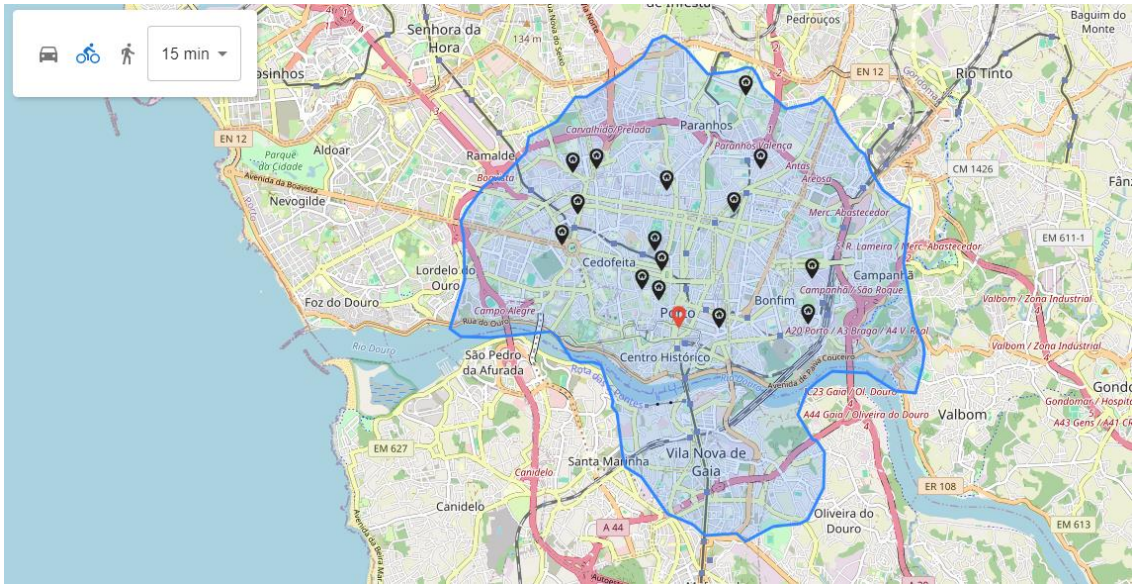

Figure 11 - Isochrone generated by cycling with a time of 15 minutes

The isochrone for cycling covers a moderate distance, depending on the availability of bike lanes and paths.

Walking Mode

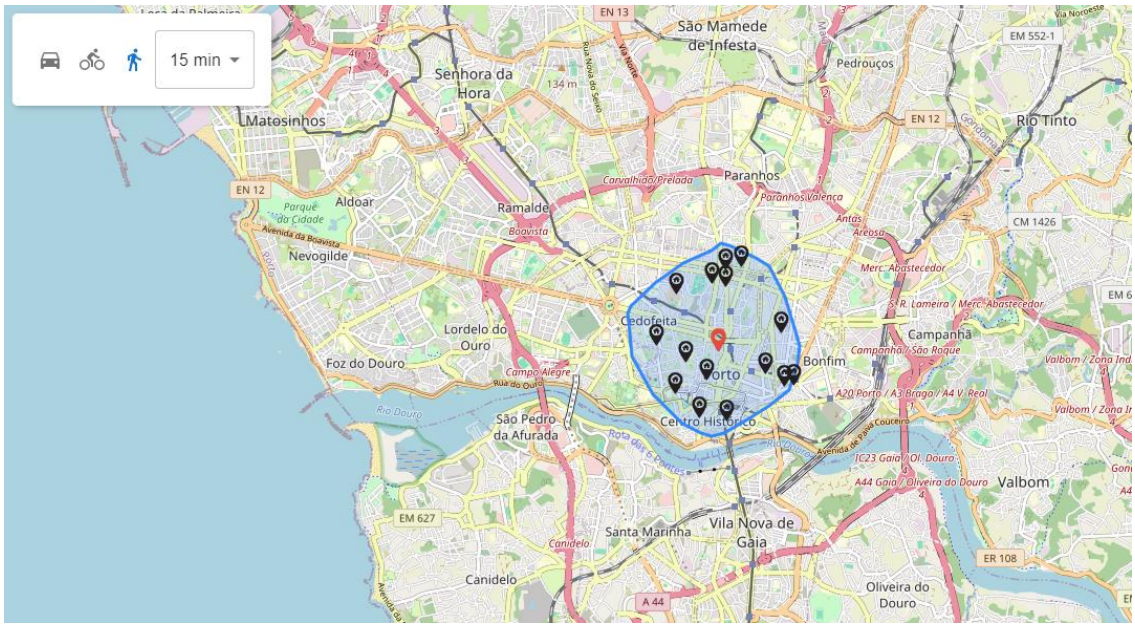

Figure 12 - Isochrone generated by walking with a time of 15 minutes

The walking isochrone covers the smallest area, reflecting slower speeds and shorter distances.

The visual comparison highlights the differences in reachability for the same travel time across the three modes, emphasizing the flexibility the system provides in catering to different user needs.

5.2.9 Isochrone Maps Comparison for Different Time frames

In addition to the comparison of travel modes, it is important to consider how the size of the isochrone changes with different time frames. The following demonstrates the isochrones generated for driving at different time intervals: 5, 15, and 30 minutes.

5 Minutes

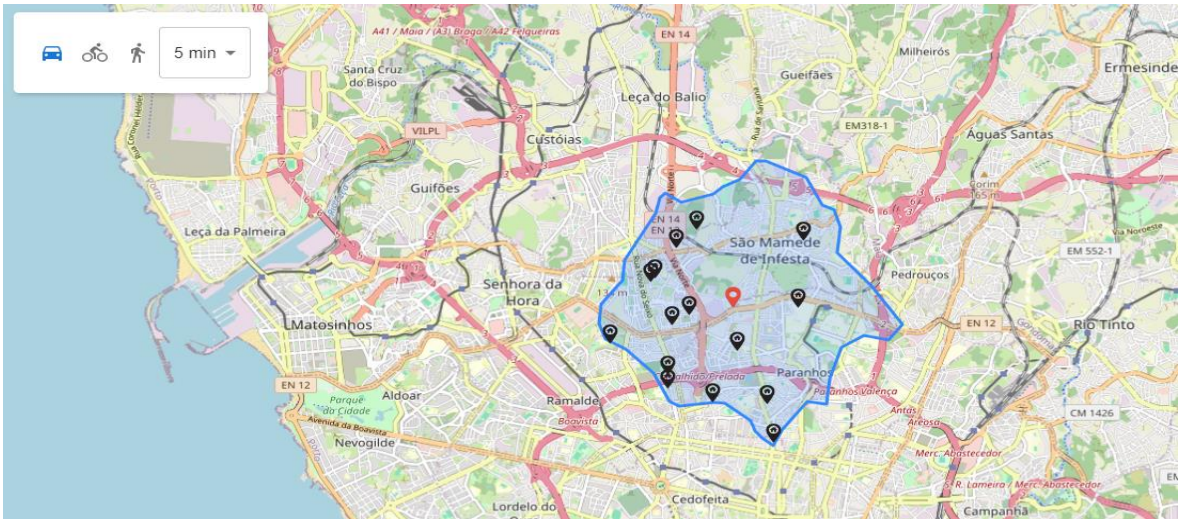

Figure 13 - Isochrone generated with a time frame of 5 minutes

15 Minutes

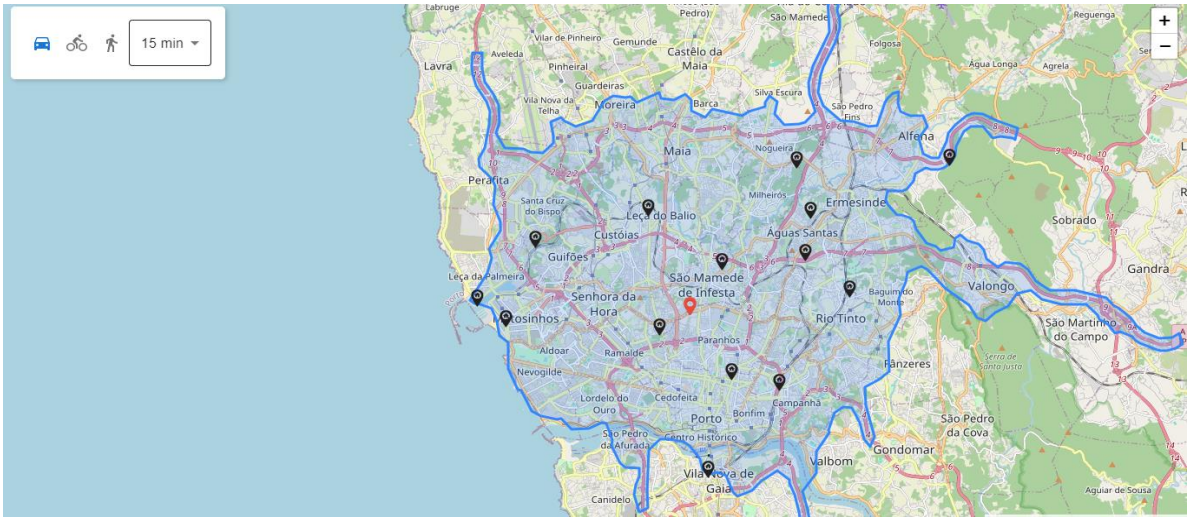

Figure 14 – Isochrone generated with a time frame of 15 minutes

30 Minutes

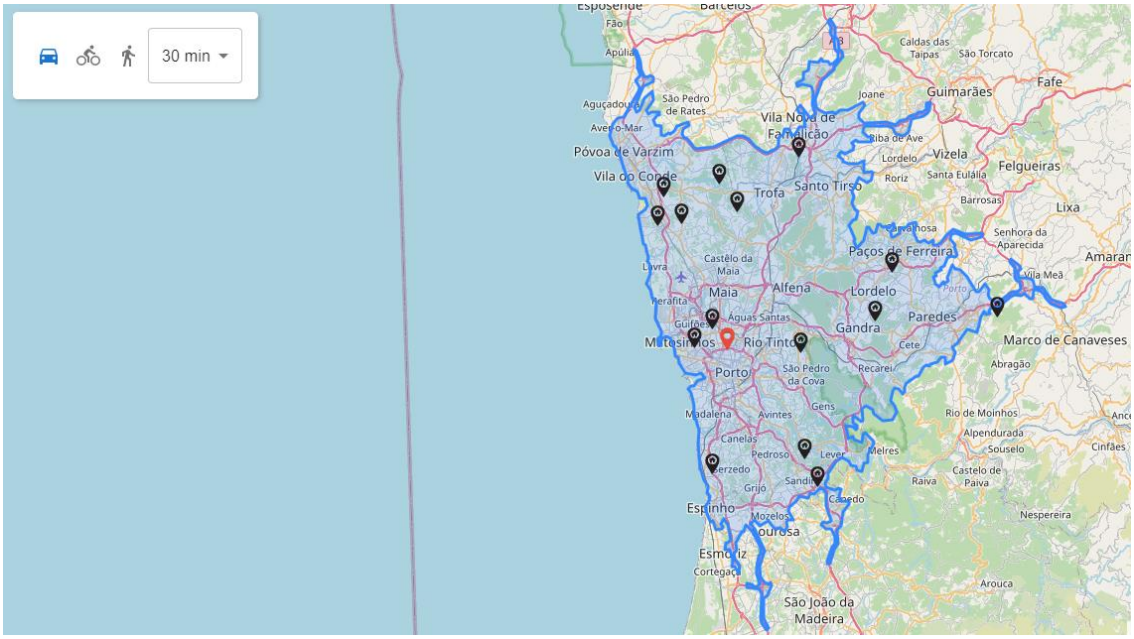

Figure 15 - Isochrone generated with a time frame of 30 minutes

In sum:

- 5 Minutes: The isochrone is small, covering areas that are very close to the starting point.
- 15 Minutes: The isochrone grows significantly, covering a broader range of properties.

- 30 Minutes: This time frame covers the largest area.

This comparison highlights the relationship between time and accessibility, allowing users to better understand how adjusting their preferred travel time impacts the range of properties available to them.

5.3 Testing and Validation

Ensuring the reliability and accuracy of the system was a critical aspect of the implementation process. Testing was conducted at both the frontend and backend levels to validate the functionality of the system and ensure that it met the requirements defined in the user stories.

5.3.1 Backend Testing

The backend, powered by ORS and OpenStreetMap data, was tested to verify that it could handle various routing requests and generate accurate isochrone maps. This was done by putting the API endpoints through various test scenarios with the different parameters for instance, testing the system's capability to handle long travel hours or areas that are hard to reach. Further performance tests were also carried out to confirm that the system could generate isochrones within reasonable time even if the system was under load.

Postman was particularly useful in testing the system's ability to handle diverse parameters, including long travel hours and areas that are hard to reach. The tool allowed for thorough validation of the API endpoints, ensuring that the system could respond efficiently to requests across different conditions. The API was tested with different parameters such as mode of transport (driving, cycling, walking) and time frame (5, 15 and 30 minutes) to validate the system's response under varying conditions, as shown in figure 17.

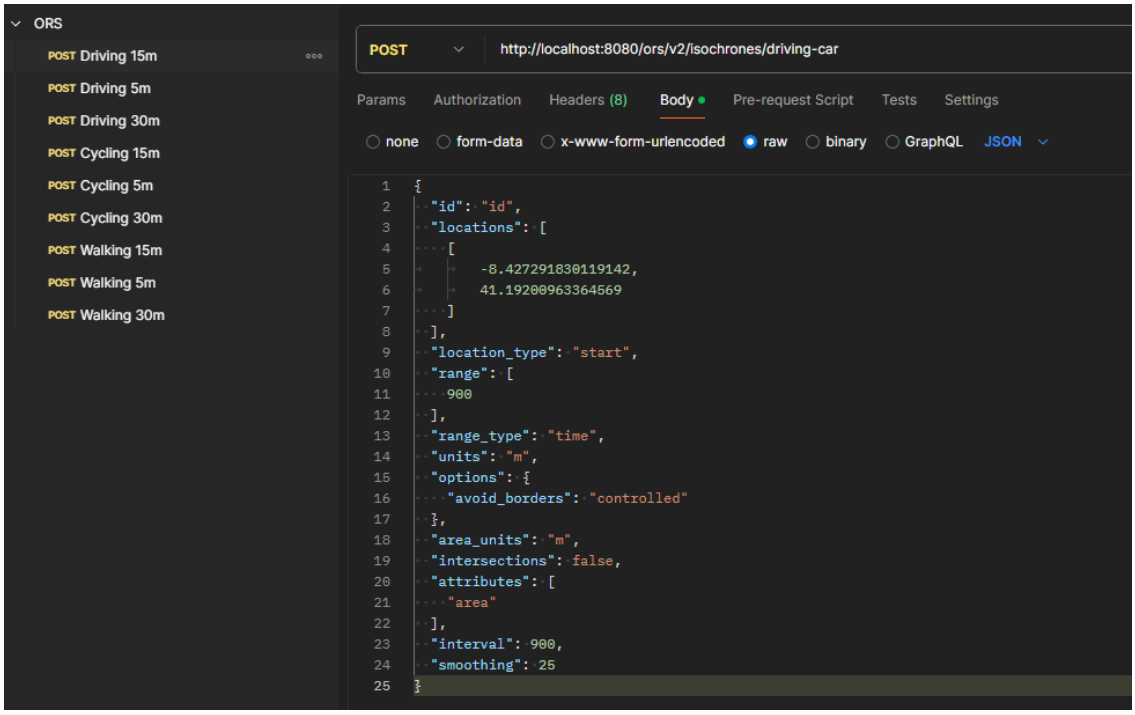

Figure 16 – Postman Test Request

To assess how well the system handled high volume of users, Apache JMeter was utilized, a widely recognized and powerful performance testing tool used for simulating user load on web applications and APIs (Apache JMeter - Apache JMeterTM, 2024), to simulate 5000 concurrent users (threads) accessing the system. The test configuration included:

- **Ramp-up period:** 1 second.
- **Loop count:** Infinite.

The response time results (Figure 18) reveal several key insights:

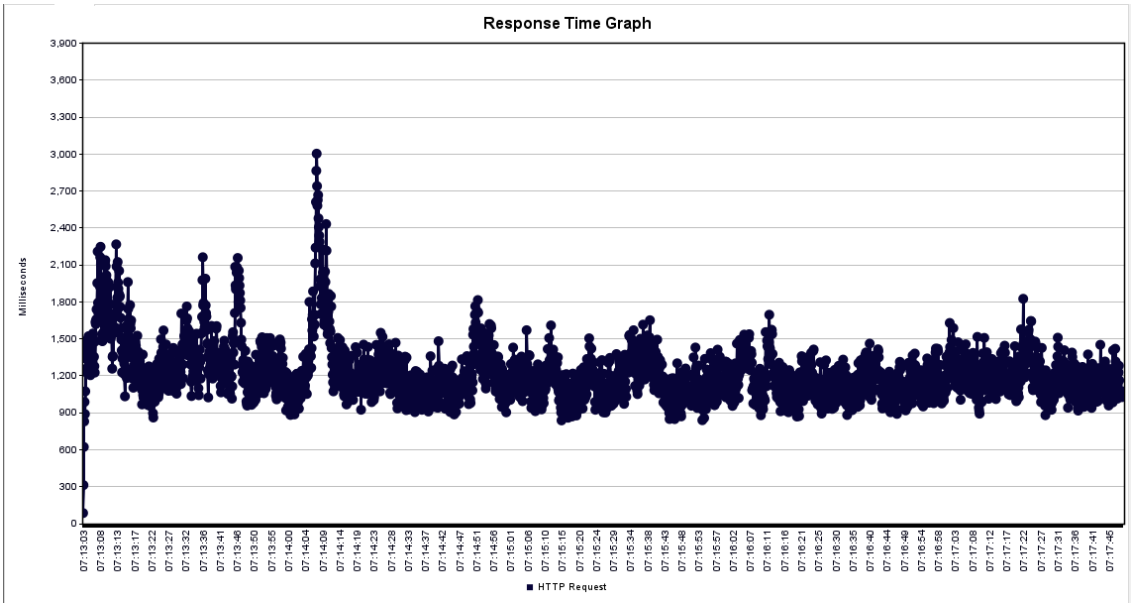

Figure 17 – ORS Response Time Graph

Initially, there was a significant spike in response time due to the sudden influx of 1000 users, with response times exceeding 2000 ms. During the test the system became more stable and the response time of most requests ranged between 800 and 1500 ms. There were some variations, which were most prominent during the peak points, where response time was sometimes higher, which could be suggestive of stress points in the server or network lag.

These results show that the system performs quite well under normal load but there is congestion when tested under high load. The 2 second response time for isochrone generation is reasonable but there are opportunities to further optimize for more stability under high loads.

React and Leaflet efficiently rendered isochrone maps and displayed filtered property results. The map rendering performance was optimized for user interactivity, with minimal lag when zooming, panning, or generating new isochrones based on user input.

5.3.2 Frontend Testing

In the frontend testing, the primary goal was to ensure that the interactions were being handled as expected while testing the UI. Some functional tests were conducted to see how the components behave when they are integrated with the ORS API. Unit tests focused on individual components, while functional tests ensured that the system as a whole behaved correctly when integrated with the ORS API. The following key aspects were tested:

- State Management: Check whether component's internal state was updated properly according to the user actions like choosing travel modes and time period.
- UI Interactions: The components were tested to check how they respond to inputs that a user might perform, including clicking on the map and changing options.
- Map and Marker Updates: Checked whether the map and markers changed according to the input of the user.
- Loading States: Ensure that loading indicators were correctly displayed when the API was being accessed and that they vanished after the response was obtained.

The following are some specific tests that were written and executed:

```
test('renders without crashing and shows map controls', () => {
    render(<IsochroneMap center={center} />);

    expect(screen.getByRole('button', { name: /driving/i
})).toBeInTheDocument();
    expect(screen.getByRole('button', { name: /walking/i
})).toBeInTheDocument();
    expect(screen.getByRole('button', { name: /cycling/i
})).toBeInTheDocument();
});
```

Code Snippet 13 - Rendering the Component with Initial Controls Test

This test confirms that the component is rendered with the right map controls for travel mode and time interval selection.

```
test('changes mode when icon buttons are clicked', () => {
    render(<IsochroneMap center={center} />);

    // Simulate mode change by clicking on the different travel
mode buttons
    const drivingButton = screen.getByRole('button', { name:
/driving/i });
    fireEvent.click(drivingButton);

    expect(drivingButton).toHaveAttribute('color', 'primary'); //
The driving mode is selected

    const walkingButton = screen.getByRole('button', { name:
/walking/i });
    fireEvent.click(walkingButton);

    expect(walkingButton).toHaveAttribute('color', 'primary'); //
The walking mode is selected
});
```

Code Snippet 14 - Handling the Mode Change Test

This test guarantees that, when the user changes the travel modes, the UI changes appropriately, and the ORS API receives the correct mode.

```
test('shows loading spinner when isochrone data is being fetched',
async () => {
    render(<IsochroneMap center={center} />);


expect(screen.queryByRole('progressbar')).not.toBeInTheDocument();

    fireEvent.click(screen.getByRole('button', { name: /driving/i
}));

    expect(screen.getByRole('progressbar')).toBeInTheDocument();

    // Wait for the API response and check spinner disappearance
    await waitFor(() => {

expect(screen.queryByRole('progressbar')).not.toBeInTheDocument();
    });
});
```

Code Snippet 15 - Display Loading Spinner During API Call Test

This test ensures that the loading spinner is active during the loading of the isochrone data and is not active once the API call request is made.

```
test('updates map with isochrone polygon and random points', async
() => {
    render(<IsochroneMap center={center} />);

    const mockResponse = {
        features: [
            {
                geometry: {
                    coordinates: [
                        [
                            [-8.427291, 41.192009],  [-8.427291,
41.592009],  [-8.827291, 41.192009],  [-8.527291, 41.992009]
                        ]
                    ]
                }
            }
        ]
    };
    (MapService.isochroneInformation as
jest.Mock).mockResolvedValue(mockResponse);

    fireEvent.click(screen.getByRole('button', { name: /driving/i
}));

    await waitFor(() => {
        const polygonElement = document.querySelectorAll('.leaflet-
interactive');
        expect(polygonElement.length).toBeGreaterThan(0);
    });

    await waitFor(() => {
        const markerElements = document.querySelectorAll('.leaflet-
marker-icon');
        expect(markerElements.length).toBe(15);
    });
});
```

Code Snippet 16 - Update Map and Markers Based on API Response Test

This test checks the functionality of the application when the API responds with isochrone data, the polygon is drawn on the map and the random markers inside the polygon are drawn correctly.

From the frontend testing, it was possible to identify the proper functionality of the user interactions, state management, and the API integration. The unit and functional tests gave confidence to the expected correctness of the UI particularly when interacting with the map and when making calls to the ORS API.

5.3.3 Usability Testing

The usability testing was done in order to evaluate the user experience and the problems that could be encountered with the interface. This involved getting feedback from the test users where the users were requested to perform some tasks such as choosing a location on the map and generate isochrones. The feedback was incorporated to the interface to make it as simple as possible for the users to navigate.

5.4 Summary

Through the development of the Isochrone Map-based real estate recommendation system, OpenRouteService was successfully implemented with React frontend and the development of a functional and appealing application. The system was designed to be very flexible and scalable to accommodate the implementation of the system in different geographical regions and locations to meet the users' needs. The routing engine was able to obtain correct and up to date geographic data using OpenStreetMap data while the frontend map display was done using the Leaflet library. It was further tested and proved that the system was developed to meet the set requirements and was reliable in providing property recommendations with regards to travel time and accessibility. usability in mind with the possibility to extend and modify it according to the needs and requirements of various geographical areas. The routing engine was able to get fresh geographic data from OpenStreetMap and the frontend map rendering was done efficiently with the help of Leaflet. Several tests were conducted to verify that the system was developed to meet the set specifications and could offer property recommendations based on travel time and accessibility.

6 Solution Evaluation

This section evaluates the effectiveness of the isochrone-based real estate recommendation system in addressing the research questions and meeting the project's objectives. The evaluation was conducted through performance tests focusing on key metrics such as efficiency, and scalability.

6.1 Research questions

- **What are the specific issues of current recommendation systems for real estates concerning individuality and relevance?**

Most conventional real estate recommendation systems are based on the inherent characteristics of properties (price, size, location) while the aspect of accessibility and user uniqueness is left unaddressed. This results in the absence of customization, which has an impact on the relevance of the recommendations. It does not consider the user's lifestyle, mode of transport, and the distance to important facilities.

- **How do these limitations affect the relevance of the recommendations provided to users?**

As a result, users can be offered options that are compliant with certain accessibility requirements but do not fully fit their needs (e. g., travel time, proximity to shops and other necessary facilities). This gap between the user needs and the recommendations leads to a low satisfaction and the efficiency of the system.

- **How does the use of isochrone maps help to solve or even eliminate these limitations?**

Isochrone maps offer a solution to these issues by using travel time from a particular point. This enables the user to see which properties are available within a given time span, thus making recommendations more relevant to the property's characteristics.

- **What are the particular advantages of employing isochrone maps to improve the customization and effectiveness of the real estate recommendations?**

The major strength of isochrone maps is that it can provide recommendations that are customized according to accessibility. The ability to filter properties not only by size or price but by the availability of essential services or workplaces makes the search much more personalized and satisfying for users.

- **To what degree does the addition of isochrone maps to a recommendation system affect its effectiveness and its users' satisfaction?**

The addition of isochrone maps enhances the functionality of the system by providing property suggestions that are relevant to users' daily lives, including time to travel and accessibility to amenities.

- **In what way does the use of isochrone maps affect the general perception of users in property searches?**

Isochrone maps offer a clear and intuitive way for users to understand property accessibility in real time. Users found this visual approach easier to grasp, allowing them to better evaluate properties in relation to their commuting needs and proximity to essential services, like schools and workplaces. The ability to visualize reachable areas within a specified time frame greatly improved the decision-making process for users.

6.2 Performance Evaluation

To assess the system's performance, several tests were conducted focusing on response time and scalability.

Efficiency of the Recommendation Process

Objective: The system should generate isochrone maps and property recommendations in under 5 seconds**.**

Results: Response times were tested, with different time frames and transportation modes. Across all test scenarios, the system generated isochrones and filtered properties within an average of 1 to 5 seconds, meeting the performance requirements.

Scalability

Objective: The system should maintain acceptable performance even with increasing user traffic.

Results: Scalability was tested with JMeter simulating up to 5,000 users. The system maintained an average response time of 1-2 seconds, confirming that it scales effectively without significant performance degradation. The response time graph (Figure 17) demonstrates that, although there were occasional peaks (up to 3,000 milliseconds), the system handled the load efficiently.

7 Conclusion

This thesis aimed at solving a major gap in prior real estate recommendation systems, which are primarily based on extrinsic and intrinsic properties of the properties such as price, size, and location. These systems often fail to take into consideration such aspects as accessibility and convenience, which are gaining higher importance from the side of potential buyers, especially in the context of rapidly growing urbanization. Due to these limitations, this research introduced the use of isochrone maps, which are graphical representations of areas that can be reached within a given amount of time, to improve the depth and relevance of the offered real estate recommendations.

Based on the literature review of the current status of recommendation systems and the evaluation of isochrone maps' features, a prototype system was designed and deployed. The system developed with ORS as the backend and Leaflet as the frontend proved that travel time and spatial accessibility can be integrated into real estate searches. In contrast to the conventional systems which are rigid and based on simple spatial distance, the isochrone-based system provides the user with more flexible and tailored recommendations depending on the mode of transport and the maximum time of travel.

In conclusion, this thesis has effectively demonstrated the possibility of using isochrone maps in real estate recommendation systems. The incorporation of spatial accessibility data allows users to make better decisions by taking into account travel time and convenience factors that were not considered in the previous systems. The study implies that such an approach can dramatically change the way properties are promoted to meet the new trends in the market of properties for sale and for rent.

7.1 Future work

While this research has demonstrated the benefits of integrating isochrone maps into real estate recommendation systems, several opportunities for future development and exploration remain. Here are some points to consider in future work:

- The current prototype only includes basic travel modes of transport such as walking, cycling, and driving. Subsequent releases can extend this capability to incorporate other means of transportation such as public transport schedules.
- While the current system of calculating travel time is based on static data, the use of real time traffic information would greatly improve the usability of the system.

By addressing these areas, future research can increase the utility of isochrone-based real estate recommendation systems, ensuring that they remain flexible, scalable, and sensitive to the different needs of homebuyers and real estate agents. The continuous development of such systems has the potential to alter how people interact with real estate, making the search experience more efficient, personalized, and, ultimately, enjoyable.

References

Apache JMeter - Apache JMeter™. (n.d.). Retrieved September 14, 2024, from https://jmeter.apache.org/

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. In *Communications of the ACM* (Vol. 53, Issue 4). https://doi.org/10.1145/1721654.1721672

Bass, L., Clements, P., & Kazman, R. (2003). Software Architecture in Practice , Second Edition. In *Software Architecture*.

Baszuro, P., & Swacha, J. (2020). Requirement Engineering as a Software Development Process. *Lecture Notes on Data Engineering and Communications Technologies*, *40*, 21–39. https://doi.org/10.1007/978-3-030-34706-2_2

Choi, Y. (2023). GeoAI: Integration of Artificial Intelligence, Machine Learning, and Deep Learning with GIS. *Applied Sciences 2023, Vol. 13, Page 3895*, *13*(6), 3895. https://doi.org/10.3390/APP13063895

Choy, L. H. T., & Ho, W. K. O. (2023). The Use of Machine Learning in Real Estate Research. *Land*, *12*(4). https://doi.org/10.3390/land12040740

Church, R. L., & Cova, T. J. (2000). Mapping evacuation risk on transportation networks using a spatial optimization model. *Transportation Research Part C: Emerging Technologies*, *8*(1–6). https://doi.org/10.1016/S0968-090X(00)00019-X

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms, 4 Edition. In *The MIT Press Cambridge, Massachusetts London, England*.

Curtis, C., & Scheurer, J. (2016). PLANNING FOR PUBLIC TRANSPORT ACCESSIBILITY: An International Sourcebook. In *Planning for Public Transport Accessibility: An International Sourcebook*. https://doi.org/10.4324/9781315600758

Diário da República. (2020). *Diário da República, 2.ª série PARTE E Artigo 2.º*.

Droj, G., Kwartnik-Pruc, A., & Droj, L. (2024). A Comprehensive Overview Regarding the Impact of GIS on Property Valuation. *ISPRS International Journal of Geo-Information 2024, Vol. 13, Page 175*, *13*(6), 175. https://doi.org/10.3390/IJGI13060175

Fesenmaier, D. R., Wöber, K. W., & Werthner, H. (2006). Destination recommendation systems: Behavioural foundations and applications. In *Destination Recommendation Systems: Behavioural Foundations and Applications*. https://doi.org/10.1515/tw-2009-0214

Gentles, S. J., Charles, C., Nicholas, D. B., Ploeg, J., & McKibbon, K. A. (2016). Reviewing the research methods literature: Principles and strategies illustrated by a systematic overview of sampling in qualitative research. *Systematic Reviews*, *5*(1). https://doi.org/10.1186/s13643-016-0343-0

Geurs, K. T., & van Wee, B. (2004). Accessibility evaluation of land-use and transport strategies: Review and research directions. *Journal of Transport Geography*, *12*(2). https://doi.org/10.1016/j.jtrangeo.2003.10.005

Gharahighehi, A., Pliakos, K., & Vens, C. (2021). Recommender Systems in the Real Estate Market—A Survey. *Applied Sciences 2021, Vol. 11, Page 7502*, *11*(16), 7502. https://doi.org/10.3390/APP11167502

Google Maps Platform Documentation | Routes API | Google for Developers. (n.d.). Retrieved September 10, 2024, from https://developers.google.com/maps/documentation/routes

Gotterbarn, D., Miller, K., & Rogerson, S. (1997). Software engineering code of ethics. *Communications of the ACM*, *40*(11). https://doi.org/10.1145/265684.265699

GraphHopper Directions API with Route Optimization. (n.d.). Retrieved September 10, 2024, from https://www.graphhopper.com/

Haklay, M., & Weber, P. (2008). OpenStreetMap: User-Generated Street Maps. Pervasive Computing. *IEEE Pervasive Computing*, *7*(4).

Hoy, Z., & Xu, M. (2023). Agile Software Requirements Engineering Challenges-Solutions—A Conceptual Framework from Systematic Literature Review. *Information 2023, Vol. 14, Page 322*, *14*(6), 322. https://doi.org/10.3390/INFO14060322

Isochrone Maps: Get Real Reachability Times - Geoapify. (n.d.). Retrieved September 10, 2024, from https://www.geoapify.com/isochrone-maps/

Koç, H., Erdoğan, A. M., Barjakly, Y., & Peker, S. (2021). UML Diagrams in Software Engineering Research: A Systematic Literature Review. *Proceedings 2021, Vol. 74, Page 13*, *74*(1), 13. https://doi.org/10.3390/PROCEEDINGS2021074013

Kraus, S., Breier, M., Lim, W. M., Dabić, M., Kumar, S., Kanbach, D., Mukherjee, D., Corvello, V., Piñeiro-Chousa, J., Liguori, E., Palacios-Marqués, D., Schiavone, F., Ferraris, A., Fernandes, C., & Ferreira, J. J. (2022). Literature reviews as

independent studies: guidelines for academic practice. *Review of Managerial Science*, *16*(8), 2577–2595. https://doi.org/10.1007/s11846-022-00588-8

Leaflet - a JavaScript library for interactive maps. (n.d.). Retrieved September 10, 2024, from https://leafletjs.com/

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). Chap 1: Geographic Information : Science , Systems , and Society. *Geographic Information: Science, Systems, and Society*.

Luxen, D., & Vetter, C. (2011). Real-time routing with OpenStreetMap data. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. https://doi.org/10.1145/2093973.2094062

Mapbox. (2023). *What are Isochrones? - Mapbox*. https://www.mapbox.com/insights/isochrones

Mapbox | Maps, Navigation, Search, and Data. (n.d.). Retrieved September 10, 2024, from https://www.mapbox.com/

Neutens, T., Schwanen, T., Witlox, F., & de Maeyer, P. (2010). Equity of urban service delivery: A comparison of different accessibility measures. *Environment and Planning A*, *42*(7). https://doi.org/10.1068/a4230

openrouteservice. (n.d.). Retrieved September 10, 2024, from https://openrouteservice.org/

Páez, A., Scott, D. M., & Morency, C. (2012). Measuring accessibility: Positive and normative implementations of various accessibility indicators. *Journal of Transport Geography*, *25*. https://doi.org/10.1016/j.jtrangeo.2012.03.016

Patterson, Z., & Farber, S. (2015). Potential Path Areas and Activity Spaces in Application: A Review. *Transport Reviews*, *35*(6). https://doi.org/10.1080/01441647.2015.1042944

Zhao, Y. ;, Zhou, Y., Mu, L., Yang, J., Kainz, W., Zhao, Y., & Zhou, Y. (2024). Isochrone-Based Accessibility Analysis of Pre-Hospital Emergency Medical Facilities: A Case Study of Central Districts of Beijing. *ISPRS International Journal of Geo-Information 2024, Vol. 13, Page 288*, *13*(8), 288. https://doi.org/10.3390/IJGI13080288