COP 5536 Spring 2024

Programming Project

GatorGlide Delivery Co

**NAME:** ASNA MAHEEN

**UFID:** 68826446

**UFEMAIL:** [asnamaheen@ufl.edu](mailto:asnamaheen@ufl.edu)

# ABSTRACT

GatorGlide Delivery Co. is on a journey of extraordinary proportions. Its vision is to revolutionize the delivery logistics industry through innovation and efficiency. The company strives for operational excellence in every aspect. This abstract presents a new order management system that optimizes delivery operations using unprecedented precision and agility. This system is based on cutting-edge technology, using Adelson-Velsky-and-Landis (AVL), tree structures, as its cornerstone. It offers dynamic self-balancing architectural capabilities to quickly access and modify vast datasets.

This system is built on three foundational components. The Order Class is the first component of this system. It manages all order data, including every aspect of a delivery request. This class provides the foundation for a seamless system interaction. It includes everything from essential logistical information to customer preferences. The AVLTree class, designed for efficiency, then orchestrates the systematic organization and prioritization orders. This platform's intelligent design allows for rapid navigation and manipulation of order hierarchy, allowing for agile decision-making with live updates. OrderManagementSystem Class is designed as the central hub for delivery operations, connecting orders with AVL tree infrastructure. GatorGlide Delivery Co. offers seamless coordination and execution to ensure flawless logistics management, from order inception through fulfillment. GatorGlide Delivery Co.'s scalable architecture offers more than immediate advantages. It marks a new age of adaptability and expansion.

# INTRODUCTION

The project aims to address an inherent difficulty facing delivery companies: effectively prioritizing multiple orders in order to meet timely deliveries. GatorGlide Delivery Co. faces daily the daunting challenge of prioritizing an abundance of orders based on factors like order value, delivery time and customer preferences - but traditional methods of order management often result in inefficiency leading to delayed deliveries for some orders and processing inefficiently for others.

The purpose of this project is to design a smart order management system capable of automatically organizing orders based on priority and calculating an ETA for every delivery. By employing AVL trees--a type of self-balancing binary search tree--this project aims to equip GatorGlide Delivery Co. with tools that will quickly allow them to access and update order data. Orders of highest priority will receive priority treatment, thus minimizing risks of oversight or delay. Moreover, this system has been designed for scaleability to accommodate an increase in orders as your company expands operations. Ultimately, this project aims to streamline delivery process, reduce order management workload, and enhance overall customer satisfaction.

# SYSTEM OVERVIEW

GatorGlide Delivery Co. has implemented an order management system crafted using Python due to its user-friendliness and robust libraries, delivering an effortless yet efficient platform. At its heart lies three interlinked components which make up this order management solution:

1. Order Class: At its heart lies the Order Class, providing individual orders with their individual blueprint. This unit covers key details like ID, timer value and delivery date for optimal functioning of the system. This fundamental unit serves as the backbone of its operations.

2. AVLTree Class: Leveraging the power of an AVL tree--a self-balancing binary search tree--AVLTree Class organizes orders based on priority. This framework ensures fast order access and manipulation times amidst increasing order volumes, essential to ensure timely responses times in response to high order volumes.

3. OrderManagementSystem Class: At the core of this system lies OrderManagementSystemClass. This essential piece ensures seamless cooperation among components by overseeing order insertion, updates and cancellations as they happen using an AVALTREE to prioritize them according to priority levels and dynamically adjust ETA estimates dynamically when necessary to accommodate fluctuations.

Emphasizing modularity and scalability, the system was designed with adaptability in mind to accommodate growing order volumes efficiently. Harnessing Python and AVL trees to optimize delivery operations while increasing efficiency and reliability across the board was at the heart of this endeavor.

# IMPLEMENTATION DETAILS

GatorGlide Delivery Co.'s project unveils an advanced order management system developed in Python, chosen for its intuitive syntax and data handling abilities. Grounded in simplicity yet boasting impressive functionality, its architecture revolves around three classes named Order, AVLTree and OrderManagementSystem that each have specific roles that boost its operational prowess.

Order Class:

At the heart of order orchestration sits the Order class, serving as its core data model. This class houses critical attributes like orderId, systemTime, value, and delivery Time that represent specific details about every order. Furthermore, dynamic priority calculation uses an established formula which takes into account order value and placement time - providing an efficient means of order processing sequence optimization and thus optimizing operational efficiency.

AVLTree Class: At the heart of every successful system lies the AVLTree Class, an oasis of order management efficiency. Drawing upon its power from an AVL tree data structure, this class ensures its perpetual balance for easy operations such as insertions, deletions and searches at logarithmic time complexity. Managing orders according to priority, it engenders an orderly queue that can dynamically adapt with changing requirements - providing operational resilience despite surging order volumes.

OrderManagementSystem Class: At the heart of it all lies OrderManagementSystem,' the operational hub that drives all aspects of system functioning with precision. Integrating seamlessly the functionalities of Order and AVLTree classes, this integral component manages orders from their inception through delivery. Duties involve entering new orders into the AVL tree hierarchy on an order priority basis, recalibrating estimated times of arrival (ETAs) for all orders following any system update, and managing order cancellations with precision. Notably, dynamic recalibration of ETAs enabled by in-order traversal mechanism of an AVL tree is essential in real-time order management that keeps delivery schedule optimized according to latest order dynamics.

Implementation ethos embodies GatorGlide Delivery Co.'s dedication to code modularity and system scalability by capitalizing on Python's inherent abilities to navigate complex data structures like AVL trees with unparalleled efficiency. Intended to seamlessly adjust to fluctuating order volumes, the system lays the groundwork for GatorGlide Delivery Co. to reach unparalleled heights of effectiveness and efficiency in their delivery operations.

# PROGRAM STRUCTURE

GatorGlide Delivery Co. project's program structure was carefully created to optimize management and prioritization of delivery orders, using an AVL tree (self-balancing binary search tree) as its foundation. This approach ensures smooth operations essential to creating a balanced and organized system for handling priority delivery schedules and priorities. Below is a detailed breakdown of its core components, matching up with its code provided:

Order Class: This class represents an individual order within the system, providing all pertinent details that affect it.

As its prototype, Order(orderId, systemTime, value and deliveryTime) is its representative model; additionally its attributes include an orderId that uniquely identifies it.

-SistemulTime: This field shows when an order was first entered into the system.value is associated with each order while DeliveryTime indicates an estimated date and timeframe for its delivery.

AVLNode Class: for its primary function as the basis of an AVL tree structure. It represents individual nodes within this framework and serves as its foundational unit, called an 'AVLNode(priority).' To create this type of class: [AVLNode Class]]. [...] (Purpose]]. In fact it has three attributes which define its functionality. ([Purpose].. AVLNode Class is constructed as follows]]]. Packed into three subclasses]. It serves its primary role within an AVL tree by representing individual nodes within its hierarchy structure - thus acting as its central node component representing itself in turn representing nodes of this hierarchy structure.) [Priority)];]. (Prototype]]. (Priority). (Prototype]]. Parchete Class) [Prototype]. (AVLNode Priority). W. Inclusion] Attributes]].

Priority value assigned to nodes within an AVL tree. Essential for maintaining its order and balance.

Purpose of AVLTree Class: To facilitate all essential operations associated with an AVL tree, such as insertions, removals and traversals while maintaining its equilibrium and prioritizing orders accordingly.

Key Methods of Action:

Inject (Self, Root, Priority and Order): Inserting new orders to the tree according to priority will help maintain balance within its structure.

* Remove (self, root, priority and order): Eliminate an order or node with a given priority from the tree by specifying its order or priority and performing necessary rebalancing operations afterwards.
* in\_order\_traversal(self, root): Executes an in-order traversal of the tree, most often used for recalculating ETAs by processing orders in priority sequence.

OrderManagementSystem Class:

[Purpose]: Provides a central hub for overseeing all orders within the system by taking advantage of the functionality offered by an AVL tree to organize and prioritize efficiently.

Key Methods:

createOrder(self, orderId, systemTime, value and deliveryTime): Used to add new orders into the system according to their calculated priorities in AVL trees.

* CancelOrder(self, orderId): Removes an order from the system and AVL tree while also recalculating ETAs as necessary for remaining orders.
* "RecalculateETAs(self)": Updates the estimated time of arrival (ETAs) for all orders following any modifications to their queue, providing for accurate delivery scheduling.

GatorGlide Delivery Co. has designed an elegant program structure to effectively manage an abundance of orders, ensuring prompt and orderly deliveries are carried out. By employing an AVL tree for order prioritization, their system easily accommodates new orders while meeting strict delivery schedules without disruption.

**CONCLUSION**

GatorGlide Delivery Co. created an advanced system tailored specifically for them that used the power of AVL trees to optimize order prioritization and streamline delivery operations, significantly streamlining delivery operations while simultaneously managing an increasing order volume more easily, while simultaneously increasing overall efficiency. A noteworthy element of the project was seeing intelligent organizational strategies work their magic on complex logistical problems in real life scenarios - which proved how transformative such strategies could be in real life scenarios.

This endeavor demonstrated the significance of strategic problem-solving and thoughtful selection of tools in effectively managing operational complexities. Future enhancements could include dynamic adjustments to order priority settings, user friendliness enhancements and comprehensive data analytics and reporting features - potentially turning this system into an indispensable asset for delivery management, creating customer satisfaction while strengthening company competitive advantage within the delivery service industry.