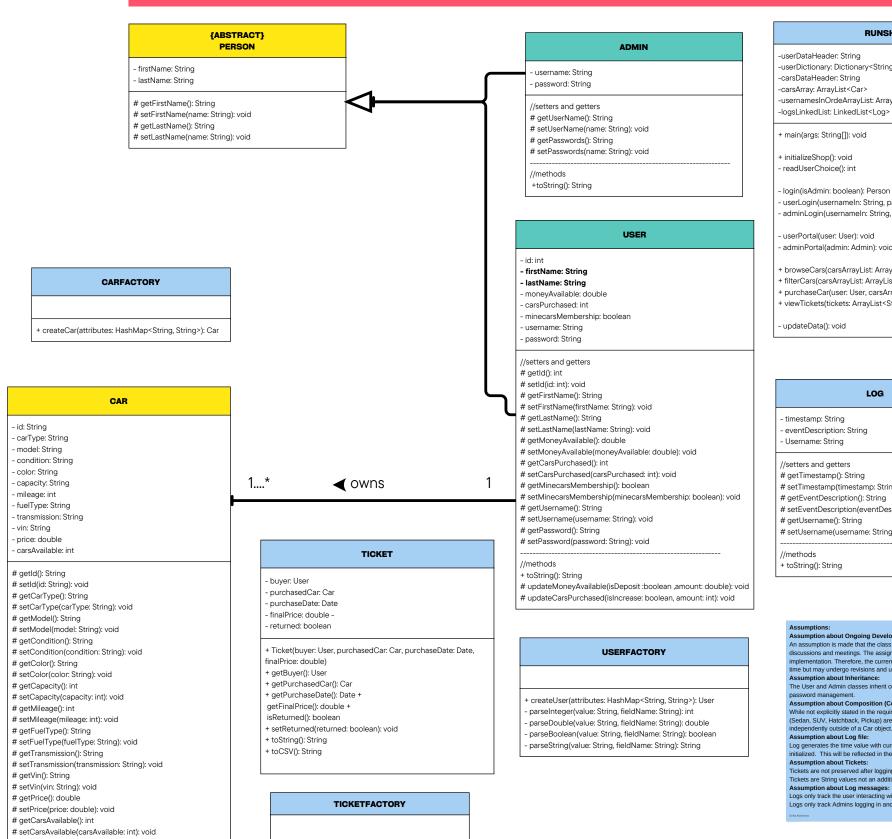
UML Class Diagram for Mine cars



//methods

+ toString(): String

updateCarsAvailable(isIncrease: boolean, amount: int): void

SHOPMANAGER

LOG

userLogin(usernameIn: String, passwordIn: String): User adminLogin(usernameln: String, passwordln: String): Admin

browseCars(carsArrayList: ArrayList<Car>): void

purchaseCar(user: User, carsArrayList: ArrayList<Car>): String

filterCars(carsArrayList: ArrayList<Car>): void

viewTickets(tickets: ArrayList<String>): void

- timestamp: String

updateData(): void

- eventDescription: String
- Username: String

userDataHeader: String

carsDataHeader: String

-carsArray: ArrayList<Car>

+ main(args: String∏): void

login(isAdmin: boolean): Person

userPortal(user: User): void adminPortal(admin: Admin): void

initializeShop(): void readUserChoice(): int

userDictionary: Dictionary<String, User>

usernamesInOrdeArrayList: ArrayList<String>

- //setters and getters
- # getTimestamp(): String
- # setTimestamp(timestamp: String): void
- # getEventDescription(): String
- # setEventDescription(eventDescription; String); void
- # getUsername(): String
- # setUsername(username: String): void
- toString(): String

FILEHANDLER

- fileHeaderIndex(filename: String): HashMap<String, Integer>
- + createCarMap(filename: String): HashMap<Integer, Car>
- createUserMap(filename: String): HashMap<Integer, User>

Assumption about Ongoing Development:

An assumption is made that the class diagram presented is a work in progress and subject to change based on team discussions and meetings. The assignment description indicates that progress is expected rather than a fully completed implementation. Therefore, the current class diagram reflects the team's understanding and progress at a specific point ir time but may undergo revisions and updates as the project evolves and requirements are refined during team meetings.

Assumption about Inheritance:
The User and Admin classes inherit common behaviors or attributes from a Person class, such as username and

Assumption about Composition (Conceptual):

While not explicitly stated in the requirements, a conceptual assumption could be made that different types of cars (Sedan, SUV, Hatchback, Pickup) are compositions of a Car object. This implies that each type of car cannot exist

independently outside of a Car object. **Assumption about Log file:**Log generates the time value with current date and time and converts it to timestamp value when an object of Log is

initialized. This will be reflected in the Log constructor which will only take two in

Assumption about Tickets:
Tickets are not preserved after logging out.
Tickets are String values not an additional class.

Assumption about Log messages:
Logs only track the user interacting with the purchase car menu when they actually purchase a car. Logs only track Admins logging in and logging out.

TESTCASES

No Attributes

 $\label{thm:carsStringsAndConstructors} \mbox{$\tt \#testCarsString>$, carsArray: ArrayList<Car>$): void}$ #testUsersStringsAndConstructors(users: ArrayList<String>, userDictionary: Dictionary<String,User>): void