In [7]:

```python
import datetime

# Class representing a library item (e.g., book, magazine, DVD)
class LibraryItem:
    def __init__(self, title, author, category, item_id):
        self.title = title
        self.author = author
        self.category = category
        self.item_id = item_id
        self.checked_out = False
        self.due_date = None

    def check_out(self, days=14):
        if not self.checked_out:
            self.checked_out = True
            self.due_date = datetime.date.today() + datetime.timedelta(days=days)
            return True
        return False

    def return_item(self):
        if self.checked_out:
            self.checked_out = False
            overdue_days = (datetime.date.today() - self.due_date).days
            self.due_date = None
            return max(overdue_days, 0)
        return -1

# Class representing the library system
class Library:
    def __init__(self):
        self.items = {}
        self.current_id = 1

    def add_item(self, title, author, category):
        self.items[self.current_id] = LibraryItem(title, author, category, self.current_id)
        print(f"Item '{title}' added with ID {self.current_id}")
        self.current_id += 1

    def check_out_item(self, item_id):
        item = self.items.get(item_id)
        if item and item.check_out():
            print(f"Item '{item.title}' checked out successfully.")
        else:
            print("Item ID not found or already checked out.")

    def return_item(self, item_id):
        item = self.items.get(item_id)
        if item:
            overdue_days = item.return_item()
            if overdue_days >= 0:
                fine = overdue_days * 0.50
                print(f"Item '{item.title}' returned successfully. Overdue fine: ${fine:.2f}" if fine > 0 else "No overdue fine.")
            else:
                print("Item was not checked out.")
        else:
            print("Item ID not found.")

    def search_items(self, search_term, search_by="title"):
        results = [item for item in self.items.values() if search_term.lower() in getattr(item, search_by).lower()]
        if results:
            for item in results:
                status = "Checked Out" if item.checked_out else "Available"
                print(f"ID: {item.item_id}, Title: {item.title}, Author: {item.author}, Category: {item.category}, Status: {status}")
        else:
            print("No items found.")

def main():
    library = Library()
    actions = {
        '1': lambda: library.add_item(input("Enter title: "), input("Enter author: "), input("Enter category: ")),
        '2': lambda: library.check_out_item(int(input("Enter item ID to check out: "))),
        '3': lambda: library.return_item(int(input("Enter item ID to return: "))),
        '4': lambda: library.search_items(input("Enter search term: "), input("Search by (title, author, category): ").lower())
    }

    while True:
        print("\nLibrary Management System\n1. Add new item\n2. Check out item\n3. Return item\n4. Search for items\n5. Exit")
        choice = input("Enter your choice: ")
        if choice == '5':
            break
        action = actions.get(choice)
        if action:
            try:
                action()
            except ValueError:
                print("Invalid input. Please enter the correct format.")
        else:
            print("Invalid choice. Please enter a number from 1 to 5.")

if __name__ == "__main__":
    main()
```

```
Library Management System
1. Add new item
2. Check out item
3. Return item
4. Search for items
5. Exit
Enter your choice: 1
Enter title: C
Enter author: Dennis
Enter category: CSE
Item 'C' added with ID 1

Library Management System
1. Add new item
2. Check out item
3. Return item
4. Search for items
5. Exit
Enter your choice: 2
Enter item ID to check out: 1
Item 'C' checked out successfully.

Library Management System
1. Add new item
2. Check out item
3. Return item
4. Search for items
5. Exit
Enter your choice: 3
Enter item ID to return: 1
No overdue fine.

Library Management System
1. Add new item
2. Check out item
3. Return item
4. Search for items
5. Exit
Enter your choice: 5
```

In [ ]:

In [ ]: