

پروپوزال پروژه پیاده‌سازی زبان خاص دامنه (DSL) برای مدیریت فروشگاه آنلاین با استفاده از ANTLR

مقدمه

هدف این پروژه پیاده‌سازی یک زبان خاص دامنه (DSL) برای مدیریت فروشگاه آنلاین با استفاده از ANTLR (نسخه ۴) است. این زبان تمرکز خود را بر روی وظایف مدیریتی در بخش پشتیبانی فروشگاه آنلاین قرار می‌دهد که به کاربران این امکان را می‌دهد تا محصولات و حساب‌های کاربری را مدیریت کنند. همچنین، خروجی این DSL به جای اینکه به طور مستقیم بر فروشگاه اثر بگذارد، کدهایی را تولید می‌کند که در زبان‌های دیگر اجرا می‌شوند و عملیات‌های مختلف را در فروشگاه آنلاین انجام می‌دهند (مانند تولید API calls یا کدهای مناسب برای ارتباط با پایگاه داده و مدیریت فروشگاه).

اهداف پروژه

- توسعه یک زبان خاص دامنه برای انجام کارهای مدیریتی در فروشگاه آنلاین.
- تعریف و پیاده‌سازی گرامر واضح و مختصر با استفاده از ANTLR ایجاد و حذف محصولات
- مدیریت حساب‌های کاربری (مثلاً افزودن، به‌روزرسانی و حذف حساب‌های کاربری)
- مدیریت موجودی محصولات (به‌روزرسانی سطح موجودی)

این DSL به کاربران اجازه خواهد داد تا این وظایف را از طریق دستورات ساده و قابل خواندن انجام دهند که توسط سیستم تجزیه و اجرا می‌شوند. همچنین، خروجی این DSL به صورت تولید کد خواهد بود که بعد از اجرا یا استفاده در محیط واقعی، عملیات‌های مورد نظر را روی فروشگاه انجام می‌دهد.

پیشینه

مدیریت یک فروشگاه آنلاین شامل وظایف تکراری بسیاری از جمله به‌روزرسانی جزئیات محصولات، مدیریت کاربران و تنظیمات موجودی محصولات است. یک زبان خاص دامنه می‌تواند این فرآیندها را با انتزاع کردن پیچیدگی‌های مربوط به کدهای خام یا API ها بسیار ساده‌تر کند.

ویژگی‌های پیشنهادی

DSL پیشنهادی تمرکز خواهد داشت بر روی ویژگی‌های زیر:

۱. ایجاد و حذف محصول:

- کاربران قادر خواهند بود محصولات جدیدی را به فروشگاه اضافه یا حذف کنند.
- دستورات:

▪ `createProduct(name, price, description, category)`

▪ `deleteProduct(productId)`

۲. مدیریت کاربر:

- ایجاد، به‌روزرسانی و حذف حساب‌های کاربری.
- دستورات:

▪ `createUser(username, password, role)`

▪ `updateUser(username, newPassword, newRole)`

▪ `deleteUser(username)`

۳. مدیریت موجودی:

- کاربران قادر خواهند بود موجودی محصولات را به‌روزرسانی کنند.
- دستورات:

▪ `updateStock(productId, newStockLevel)`

دامنه و جزئیات

۱. مدیریت محصول:

- DSL به کاربران این امکان را می‌دهد که محصولات جدیدی را با ویژگی‌های پایه‌ای (نام، قیمت، توضیحات، دسته‌بندی) اضافه کنند. حذف محصول بر اساس شناسه محصول انجام می‌شود.
- مثال دستور: `createProduct("Laptop", 1000, "High-end laptop", "Electronics")`

۲. مدیریت کاربر:

- دستورات ساده برای ایجاد کاربران با نقش‌های مختلف (مثلاً مدیر، مشتری)، به‌روزرسانی رمز عبور یا تغییر نقش و حذف کاربران.
- مثال دستور: `createUser("johnDoe", "password123", "customer")`

۳. به‌روزرسانی موجودی:

- کاربران قادر خواهند بود موجودی محصولات را به‌روزرسانی کنند. سیستم تضمین می‌کند که سطح موجودی منفی نشود.
- مثال دستور: `updateStock("Laptop", 50)`

تولید کد و ارتباط با فروشگاه آنلاین

تولید کد فرآیندی است که در آن، دستورات نوشته‌شده در DSL به کدهای برنامه‌نویسی تبدیل می‌شوند که در محیط‌های اجرایی واقعی قابل استفاده هستند. در این پروژه، خروجی DSL به جای اینکه مستقیماً عملیات‌ها را بر روی فروشگاه انجام دهد، کدهایی تولید می‌کند که در یک محیط واقعی اجرا می‌شوند تا عملیات‌های مختلف را در فروشگاه آنلاین انجام دهند. این کدهای تولیدی شامل API calls، دستورات پایگاه داده یا سایر عملیات‌های مدیریتی هستند که می‌توانند به فروشگاه آنلاین اعمال شوند.

مثال:

فرض کنید یک کاربر دستور `createProduct` را در DSL وارد می‌کند:

```
createProduct("Laptop", 1000, "High-end laptop", "Electronics")
```

در اینجا، DSL به جای اینکه مستقیماً اطلاعات را در پایگاه داده فروشگاه وارد کند، یک کد تولید می‌کند که به صورت زیر در زبان برنامه‌نویسی Python می‌تواند اجرا شود:

```
import requests

url = "https://example.com/api/products"
payload = {
    "name": "Laptop",
    "price": 1000,
    "description": "High-end laptop",
    "category": "Electronics"
}
response = requests.post(url, json=payload)

if response.status_code == 201:
    print("Product created successfully!")
else:
    print("Error creating product.")
```

در این مثال، کد تولیدی یک `API call` است که اطلاعات محصول را به سرور ارسال می‌کند تا آن را به پایگاه داده فروشگاه اضافه کند.

این درخواست به سرور ارسال می‌شود تا یک محصول جدید به سیستم فروشگاه آنلاین اضافه کند. در اینجا، زبان خاص دامنه شما تنها کد API را تولید کرده و مسئولیت اجرای آن با سیستم خارجی (مانند سرور یا پایگاه داده) است.

```
POST /api/products
Content-Type: application/json
{
  "name": "Laptop",
  "price": 1000,
  "description": "High-end laptop",
  "category": "Electronics"
}
```

مراحل و زمان‌بندی پروژه

۱. مرحله اول: طراحی گرامر و پیاده‌سازی Lexer

- تعریف گرامر DSL برای مدیریت محصول، مدیریت کاربر و به‌روزرسانی موجودی محصولات.
- پیاده‌سازی Lexer و تجزیه‌گر با استفاده از ANTLR.

۲. مرحله دوم: ساخت درخت نحو و تفسیر دستورات

- ساخت درخت نحو انتزاعی (AST) از ورودی تجزیه شده.
- پیاده‌سازی اجرای دستورات برای هر ویژگی.
- در این مرحله، خروجی DSL به کدهایی تولید خواهد شد که در زبان‌های برنامه‌نویسی مانند Python می‌توانند اجرا شوند و عملیات‌های مختلف را در فروشگاه انجام دهند. این کدها می‌توانند شامل API calls یا کدهای ارتباط با پایگاه داده باشند.

۳. مرحله سوم: آزمایش و اصلاحات

- آزمایش دستورات DSL برای صحت و اطمینان از اینکه تجزیه‌گر موارد لبه‌ای را به درستی مدیریت می‌کند.
- اصلاح پیاده‌سازی بر اساس بازخورد آزمایش.

نتیجه‌گیری:

به‌طور خلاصه، DSL به‌جای اینکه مستقیماً فروشگاه آنلاین را مدیریت کند، کدهایی تولید می‌کند که در python می‌تواند اجرا بشود و عملیات‌های مورد نظر را در فروشگاه آنلاین انجام دهد. این کدها می‌توانند شامل **API calls**، **دستورات پایگاه داده** یا سایر کدهای اجرایی باشند که برای تعامل با سیستم‌های مختلف فروشگاه طراحی شده‌اند.