Proposal: Migration of CopilotW Java Application to Python

Overview

This document proposes the migration of the existing CopilotW application from Java (Spring Boot) to Python. The migration aims to modernize the technology stack, improve maintainability, and leverage Python's rich ecosystem for web development.

Current Java Project Structure

```
src/
  main/
    java/
      com/copilotw/
        CopilotWApplication.java
        controllers/
          DeptController.java
          EmployeeController.java
        models/
          Employee.java
        repositories/
          EmployeeRepository.java
        services/
          EmployeeService.java
    resources/
      application.properties
  test/
    java/
      com/copilotw/
        CopilotWApplicationTests.java
```

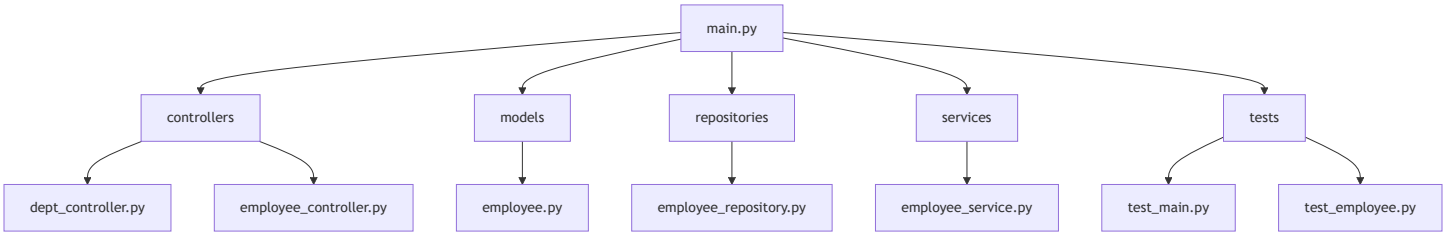
Proposed Python Project Structure

```
copilotw/
  app/
    __init__.py
    main.py
    controllers/
      dept_controller.py
      employee_controller.py
    models/
      employee.py
    repositories/
      employee_repository.py
    services/
      employee_service.py
  tests/
    test_main.py
    test_employee.py
requirements.txt
README.md
```

Migration Steps

1. **Set up Python environment** (recommend using `venv` or `conda`).
2. **Select a web framework** (Flask or FastAPI recommended for similar structure).
3. **Recreate models, controllers, services, and repositories** in Python, mirroring the Java structure.
4. **Migrate business logic** from Java to Python.
5. **Implement database integration** using SQLAlchemy or an ORM suitable for the chosen framework.
6. **Write unit tests** for all modules.
7. **Update documentation** and provide migration support.

Mermaid Diagram: Proposed Python Structure



Benefits of Migration

- Improved maintainability and readability
- Access to Python's extensive libraries
- Faster development cycles
- Easier onboarding for new developers

Conclusion

Migrating to Python will future-proof the CopilotW application and provide a robust foundation for further enhancements.