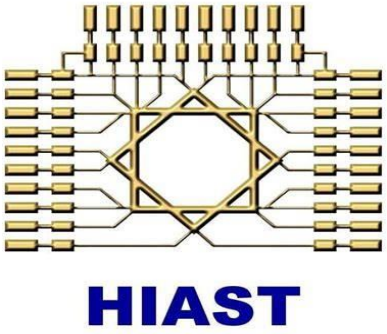
الجمعورية العربية السورية
المعهد العالي للعلوم التطبيقية و التكنولوجيا
السنة الخامسة –هندسة الشبكات ونظم التشغيل.



مقرر أساسيات في النظم الموزعة .

الوظيفة الثالثة .

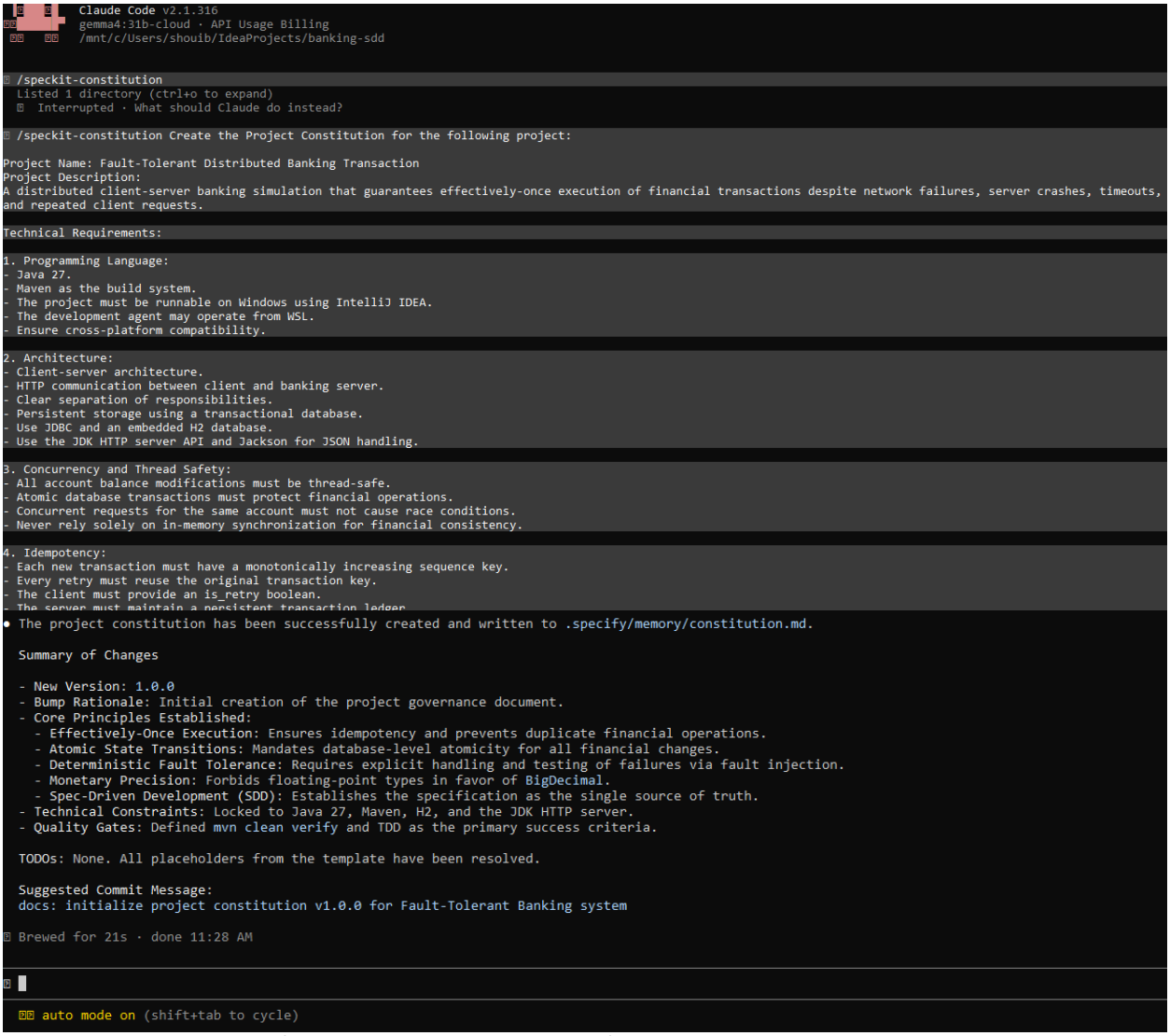
تنفيذ معاملة بنكية موزعة عالية الوثوقية

تقدمة الطالب :

شعيب صبح

منهجية العمل المتبعة :

نعطي الوكيل أمر إنشاء الدستور :



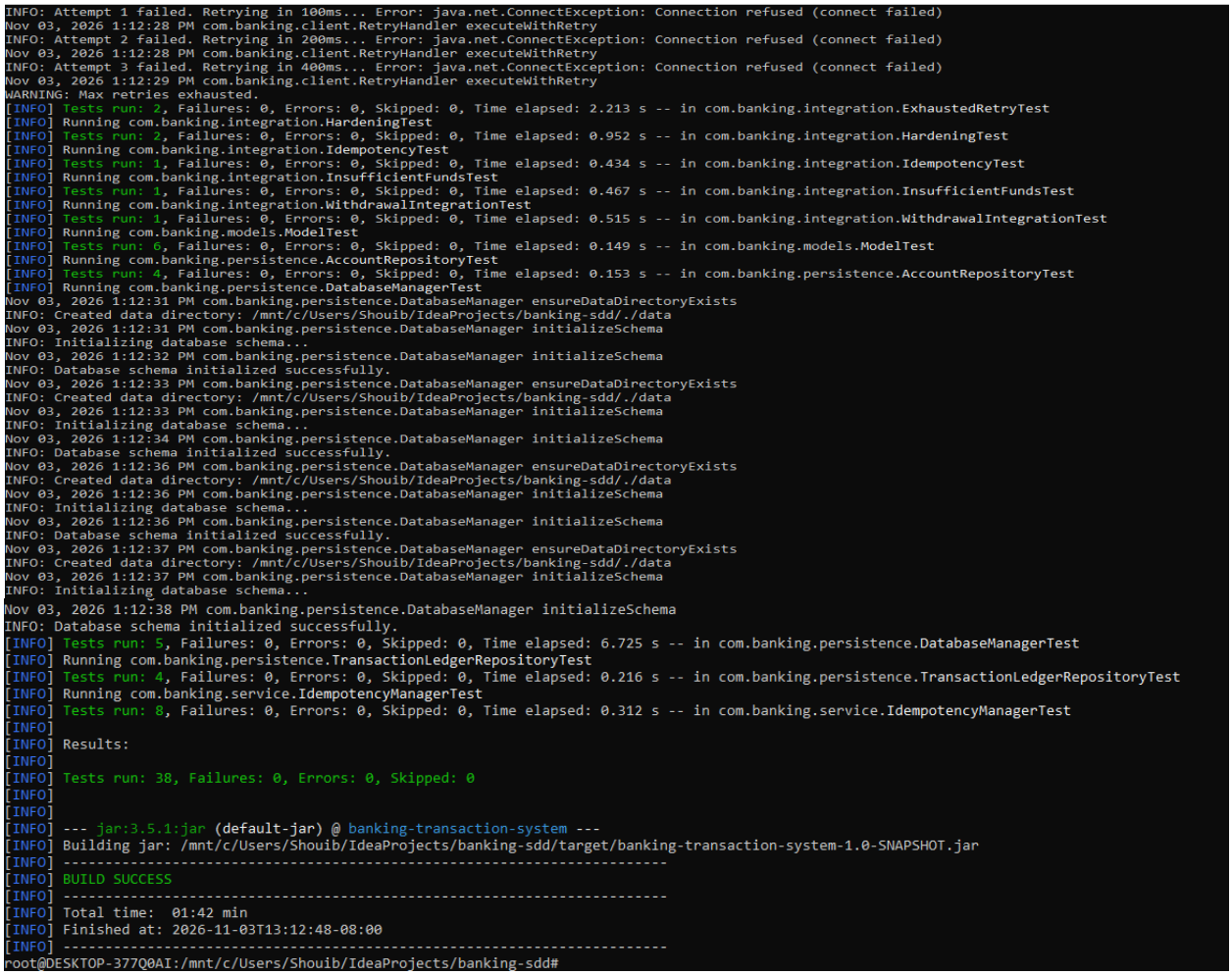
بعدما يقوم الوكيل ببناء الدستور نقوم بقرائته و التأكد منه ثم نطلب من الوكيل أي تعديلات نريدها .

بالنسبة لبقية المراحل نفس العملية نغير فقط ال skill (مثلا speckit-specify/)المستخدمة و نعطيه التوصيف الذي نريده و يطيعنا الملفات (مثل spec.md) نقوم بقرائتها و مراجعتها و نطلب من الوكيل القيام بأي تعديلات نريدها و كذلك مع ال plan و tasks و نشغل بعد كل مرحلة الامر -speckit/ clarify ليقوم الوكيل بطرح أسئلة لنا عن أي غموض أو عدم وضوح في التوصيف .

ثم نعطي الوكيل في النهاية امر speckit-implementation/ و يقوم بتنفيذ حسب التوصيف و ال plan و tasks الموجودة .

بعد الانتهاء من المشروع نشغل الاختبارات و نتأكد من صحة التنفيذ :

```
root@DESKTOP-377Q0AI:/mnt/c/Users/Shouib/IdeaProjects/banking-sdd# mvn clean verify
[INFO] Scanning for projects...
[INFO]
[INFO] ---------------< com.banking:banking-transaction-system >---------------
[INFO] Building banking-transaction-system 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] Loaded 24353 auto-discovered prefixes for remote repository central (prefixes-central.txt)
[INFO] Loaded 64 auto-discovered prefixes for remote repository apache.snapshots (prefixes-apache.snapshots.txt)
[INFO]
[INFO] --- clean:3.5.0:clean (default-clean) @ banking-transaction-system ---
[INFO] Deleting /mnt/c/Users/Shouib/IdeaProjects/banking-sdd/target
[INFO]
[INFO] --- resources:3.5.0:resources (default-resources) @ banking-transaction-system ---
[INFO] Copying 11 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ banking-transaction-system ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 18 source files with javac [debug target 27] to target/classes
[INFO] /mnt/c/Users/Shouib/IdeaProjects/banking-sdd/src/main/java/com/banking/server/RequestHandler.java: /mnt/c/Users/Shouib/IdeaProjects/banking-sdd/src/main/java/com
/banking/server/RequestHandler.java uses unchecked or unsafe operations.
[INFO] /mnt/c/Users/Shouib/IdeaProjects/banking-sdd/src/main/java/com/banking/server/RequestHandler.java: Recompile with -Xlint:unchecked for details.
[INFO]
[INFO] --- resources:3.5.0:testResources (default-testResources) @ banking-transaction-system ---
[INFO] skip non existing resourceDirectory /mnt/c/Users/Shouib/IdeaProjects/banking-sdd/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ banking-transaction-system ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 14 source files with javac [debug target 27] to target/test-classes
[INFO]
[INFO] --- surefire:3.6.0:test (default-test) @ banking-transaction-system ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.banking.concurrency.ConcurrentWithdrawalTest
Nov 03, 2026 1:12:02 PM com.banking.persistence.DatabaseManager initializeSchema
INFO: Initializing database schema...
Nov 03, 2026 1:12:09 PM com.banking.persistence.DatabaseManager initializeSchema
INFO: Database schema initialized successfully.
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 10.07 s -- in com.banking.concurrency.ConcurrentWithdrawalTest
[INFO] Running com.banking.fault.ResponseLossTest
Nov 03, 2026 1:12:16 PM com.banking.server.HttpServerApp start
INFO: Banking Server started on port 8080
Nov 03, 2026 1:12:23 PM com.banking.server.HttpServerApp stop
INFO: Banking Server stopped.
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 11.48 s -- in com.banking.fault.ResponseLossTest
[INFO] Running com.banking.fault.ServerCrashTest
Nov 03, 2026 1:12:24 PM com.banking.server.HttpServerApp start
INFO: Banking Server started on port 8080
Nov 03, 2026 1:12:24 PM com.banking.server.RequestHandler handleWithdrawal
WARNING: FaultInjector: Simulating response loss. Closing connection without responding.
Nov 03, 2026 1:12:25 PM com.banking.server.HttpServerApp stop
INFO: Banking Server stopped.
Nov 03, 2026 1:12:25 PM com.banking.server.HttpServerApp start
INFO: Banking Server started on port 8080
Nov 03, 2026 1:12:26 PM com.banking.server.HttpServerApp stop
INFO: Banking Server stopped.
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.469 s -- in com.banking.fault.ServerCrashTest
[INFO] Running com.banking.integration.ConflictTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.500 s -- in com.banking.integration.ConflictTest
[INFO] Running com.banking.integration.ExhaustedRetryTest
Nov 03, 2026 1:12:27 PM com.banking.client.RetryHandler executeWithRetry
INFO: Attempt 1 failed. Retrying in 100ms... Error: java.net.ConnectException: Connection refused (connect failed)
Nov 03, 2026 1:12:26 PM com.banking.client.RetryHandler executeWithRetry
INFO: Attempt 2 failed. Retrying in 200ms... Error: java.net.ConnectException: Connection refused (connect failed)
Nov 03, 2026 1:12:26 PM com.banking.client.RetryHandler executeWithRetry
INFO: Attempt 3 failed. Retrying in 400ms... Error: java.net.ConnectException: Connection refused (connect failed)
Nov 03, 2026 1:12:27 PM com.banking.client.RetryHandler executeWithRetry
WARNING: Max retries exhausted.
Nov 03, 2026 1:12:27 PM com.banking.server.HttpServerApp start
INFO: Banking Server started on port 8080
Nov 03, 2026 1:12:27 PM com.banking.server.HttpServerApp stop
INFO: Banking Server stopped.
Nov 03, 2026 1:12:28 PM com.banking.client.RetryHandler executeWithRetry
```



شرح بعض آليات النظام والوظائف التي تم اختبارها

1. المعالجة المتزامنة(Concurrency)

تتيح المعالجة المتزامنة استقبال عدة طلبات للسحب من الحسابات في الوقت نفسه. وتكمن أهمية ذلك في ضمان التعامل الصحيح مع الطلبات المتزامنة، بحيث لا تؤدي العمليات المتعددة إلى نتائج غير صحيحة في رصيد الحساب.

تم التحقق من ذلك باستخدام الاختبار ConcurrentWithdrawalTest، الذي يختبر سلوك النظام عند تنفيذ عمليات سحب متزامنة.

2. منع تكرار تنفيذ العمليات(Idempotency)

تُستخدم هذه الآلية لمنع تنفيذ المعاملة المالية نفسها أكثر من مرة، خصوصاً عندما لا يتلقى العميل استجابة من الخادم ويضطر إلى إعادة إرسال الطلب.

يعتمد هذا المفهوم على التعرف إلى الطلبات السابقة ومطابقة الطلب الجديد معها، مما يساعد على تجنب تكرار الخصم من الحساب.

تم اختبار هذه الوظيفة من خلال:

- IdempotencyTest
- IdempotencyManagerTest

3. آلية إعادة المحاولة(Retry Mechanism)

تسمح هذه الآلية للعميل بإعادة إرسال الطلب عند حدوث فشل مؤقت في الاتصال بالخادم، بدلاً من اعتبار العملية فاشلة مباشرة.

يستخدم النظام أسلوب التراجع الأسي(Exponential Backoff) ، حيث تزداد مدة الانتظار بين المحاولات تدريجياً. وقد ظهرت في سجل التنفيذ القيم التالية:

المحاولة	مدة الانتظار
الأولى	100 ms
الثانية	200 ms
الثالثة	400 ms

وعند استنفاد عدد المحاولات المسموح به، يتوقف النظام عن إعادة المحاولة ويصدر رسالة Max retries exhausted.

تم التحقق من هذه الآلية باستخدام الاختبار ExhaustedRetryTest.

4. التعامل مع فقدان الاستجابة(Response Loss)

قد ينجح الخادم في معالجة طلب السحب وتحديث البيانات، لكن تنقطع الاستجابة قبل وصولها إلى العميل. في هذه الحالة لا يستطيع العميل معرفة ما إذا كانت العملية قد نُفذت بالفعل أم لا.

يتحقق الاختبار ResponseLossTestمن قدرة النظام على التعامل مع هذه الحالة، كما تساعد آلية منع تكرار العمليات على تجنب تنفيذ السحب مرة أخرى عند إعادة إرسال الطلب.

5. التعامل مع تعطل الخادم(Server Crash)

يختبر هذا الجانب قدرة النظام على التعامل مع توقف الخادم ثم إعادة تشغيله، والتحقق من إمكانية متابعة العمل بعد عودة الخدمة.

تم استخدام ServerCrashTestلمحاكاة هذه الحالة، وأظهرت النتائج نجاح الاختبار، مع ظهور رسائل توضح إيقاف الخادم وإعادة تشغيله.

6. التحقق من الرصيد والتعارض بين العمليات

يتضمن المشروع اختبارات للتحقق من صحة العمليات المالية في حالات مختلفة، ومنها:

- InsufficientFundsTest: التحقق من رفض عملية السحب عند عدم كفاية الرصيد.
- ConflictTest:اختبار التعامل مع حالات تعارض العمليات.
- WithdrawalIntegrationTest: التحقق من تكامل عملية السحب بين مكونات النظام.

7. اختبارات وحدات النظام والتخزين

إلى جانب اختبارات التكامل والأعطال، تم تنفيذ مجموعة من اختبارات الوحدات للتحقق من صحة المكونات الداخلية للنظام.

شملت هذه الاختبارات:

- ModelTest:اختبار نماذج البيانات والكائنات المستخدمة في النظام.
- AccountRepositoryTest: اختبار عمليات الوصول إلى بيانات الحسابات.
- DatabaseManagerTest:اختبار إدارة قاعدة البيانات وتهيئة المخطط وإنشاء مجلد البيانات.
- TransactionLedgerRepositoryTest:اختبار عمليات التعامل مع سجل المعاملات.
- IdempotencyManagerTest: اختبار وظائف إدارة منع تكرار الطلبات.

تساعد هذه الاختبارات على اكتشاف الأخطاء في المكونات بصورة مستقلة قبل الاعتماد على اختبارات التكامل التي تتحقق من عمل المكونات معاً.

من خلال نتائج الاختبارات يمكن ملاحظة أن المشروع اجتاز جميع الاختبارات الـ 38 دون أي إخفاق، بما في ذلك اختبارات المعالجة المتزامنة، ومنع تكرار العمليات، والتعامل مع فقدان الاستجابة، وتعطل الخادم، وفشل الاتصال.

كما أظهرت الاختبارات أن آلية إعادة المحاولة تعمل وفق أسلوب التراجع الأسي، وأن النظام يستطيع اكتشاف حالات فشل الاتصال والتوقف عن المحاولة بعد استنفاد الحد الأقصى المحدد.

وبناءً على ذلك، يمكن اعتبار نتائج الاختبارات دليلاً على نجاح تنفيذ السيناريوهات التي تم اختبارها، وعلى سلامة الوظائف التي تغطيها هذه الاختبارات.