Bank Application

Dan

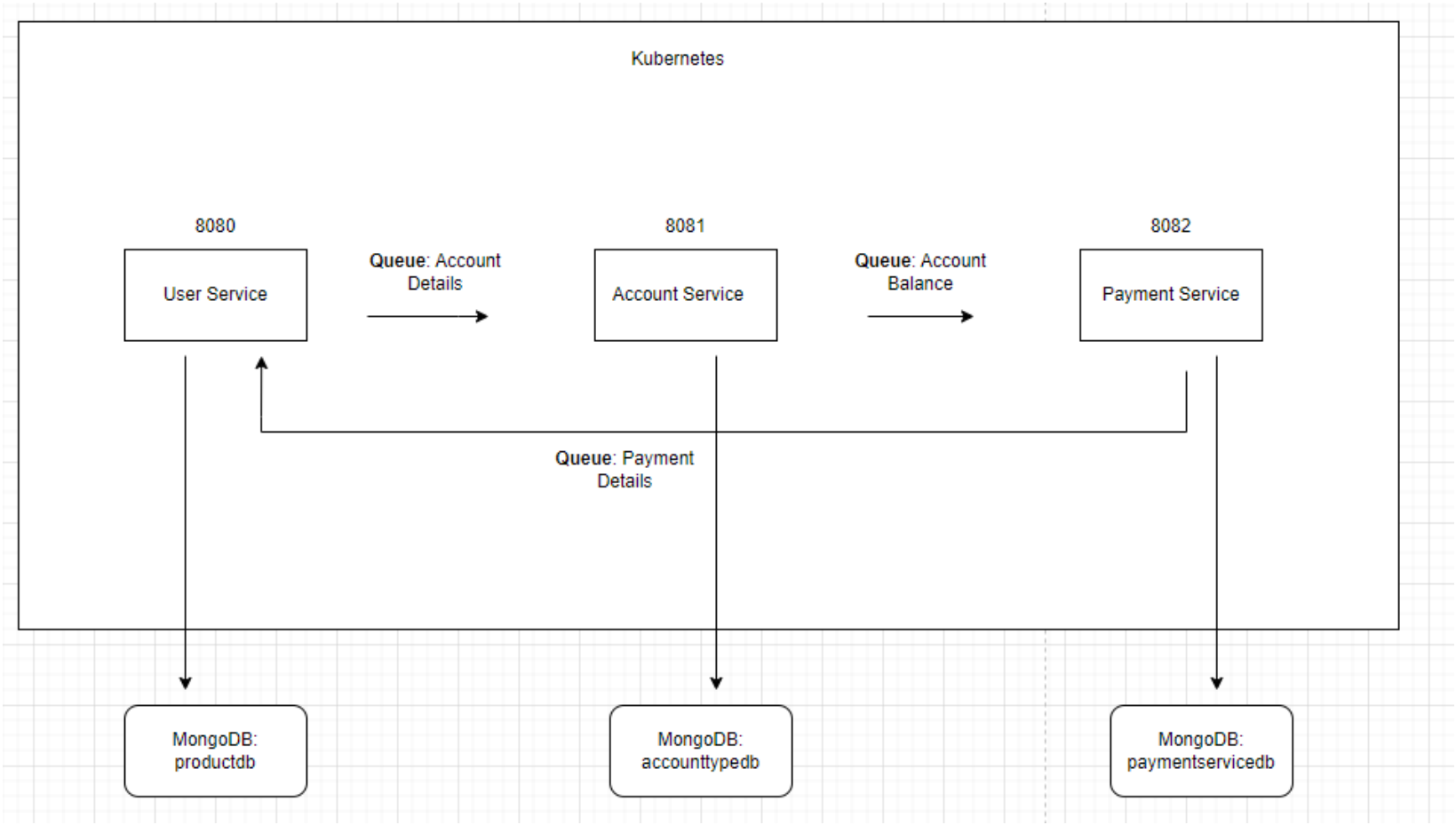
Kenneth

Andrew

Introduction

- Three Microservices
  - User Service
  - Account Service
  - Payment Service
- Features:
  - Account management
  - Checks balances and transactions
  - Makes transfers and bill payments
  - Database connection

Microservices Design



Technologies Used

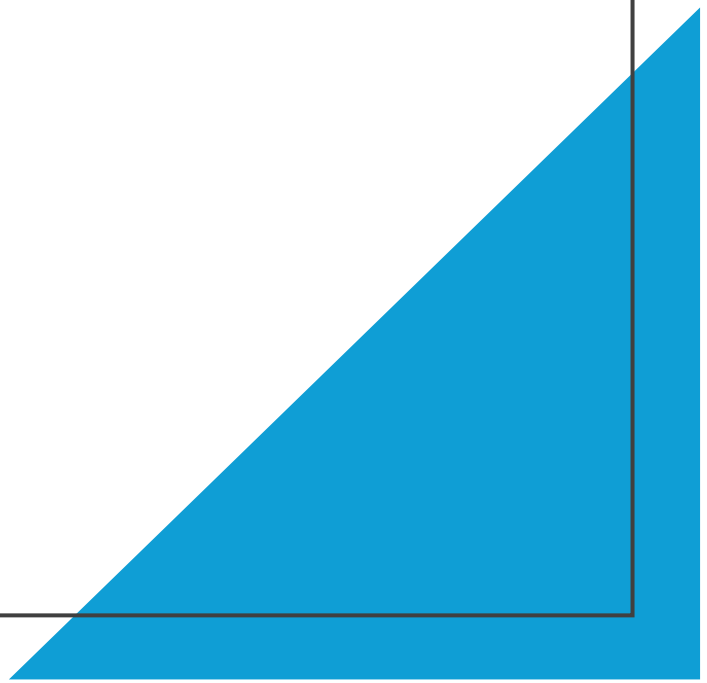
- Spring Boot
- MongoDB Atlas
- Docker
- Kubernetes
- RabbitMQ
- GitHub/GitHub Actions

CI/CD Pipeline

- Dockerfile
  - Containerize application
- GitHub Repository
- GitHub Actions Workflow File
  - Automates build and testing
  - Docker packaging and pushing
  - Deliver artifacts to repository

Testing

- Unit Testing
  - Service Classes
  - Junit and Mockito
  - Ensure internal logic function correctly
- Integration Testing
  - Controller Classes
  - MockMvc
  - Simulate HTTP requests
  - Ensure REST endpoints function correctly



Challenges and Solutions

- Message Queue Communication
  - Unsure how to integrate RabbitMQ into microservices architecture
  - **Solution**: Examined the slides on Moodle and used week 9 lab as an example.
- Reliable Integration Testing for Microservices
  - Difficulty in simulating HTTP requests
  - Challenges on how to mock dependencies
  - **Solution**:
    - Revisited Moodle slides and prior lab exercises
    - Shared findings and examples within the team
    - Made use of error feedback from Intellij terminal

Future Improvements

- Adding security features such as secure login's
- HTTPS/SSL Encryption
- Use NewSQL Distributed Database
  - For Scalability,
  - Strong consistency,
  - ACID Guarantees.