Project Demo
Stock Portfolio

Presented by Sang-Miss-Group 3
Members: Joyce Li, Meijing Li, Shehrooz Khandaker, Xinran Li

August 25, 2022

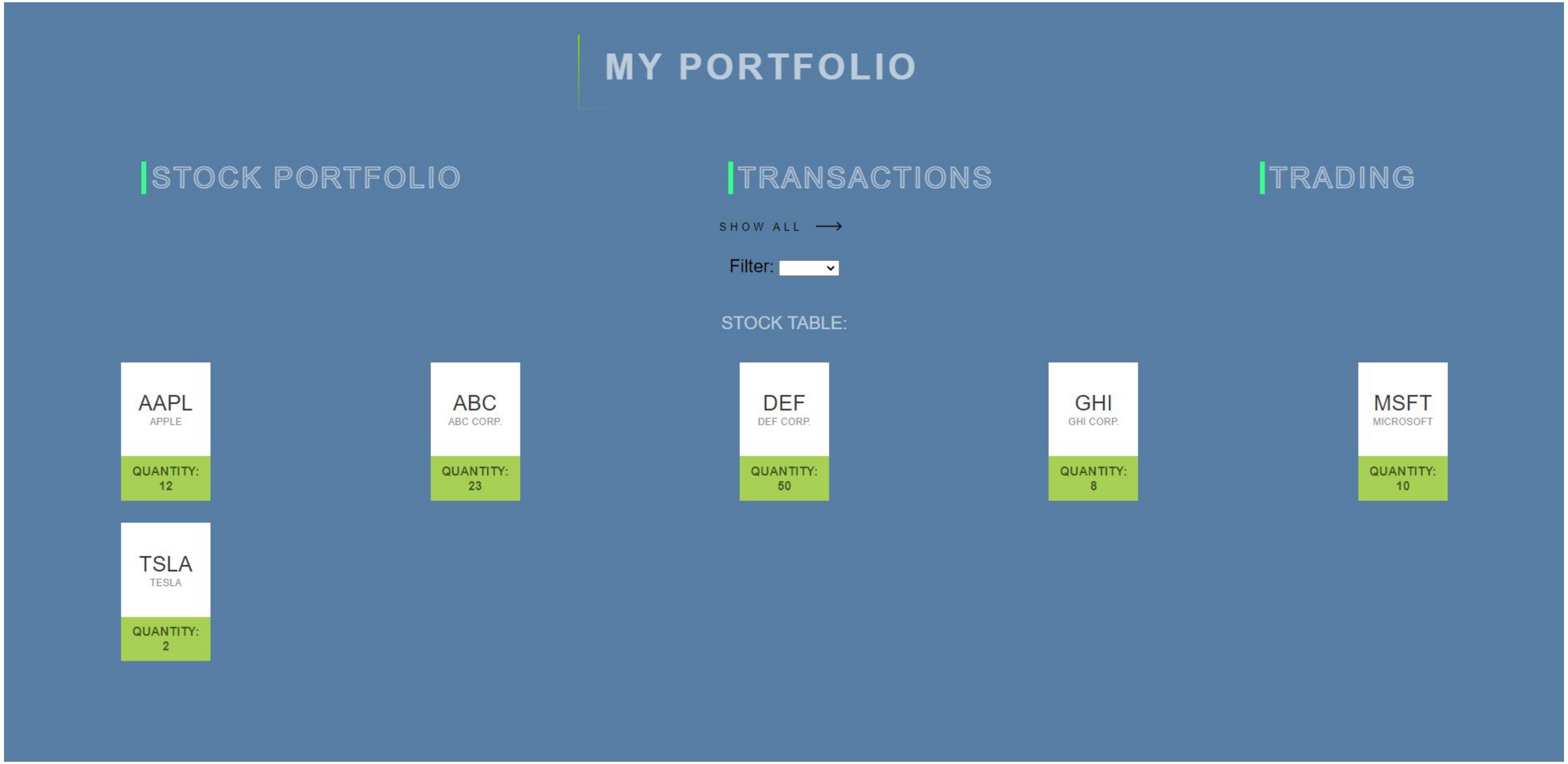
Business Need

Customers require a way to manage their investment portfolio on the website.

Technical Solution

Full-stack application for customers to view, buy, and sell their stocks.

Technical Solution - Main Page



MY PORTFOLIO

STOCK PORTFOLIO

TRANSACTIONS

TRADING

SHOW ALL ⟶

Filter:

STOCK TABLE:

AAPL
APPLE
QUANTITY: 12

ABC
ABC CORP.
QUANTITY: 23

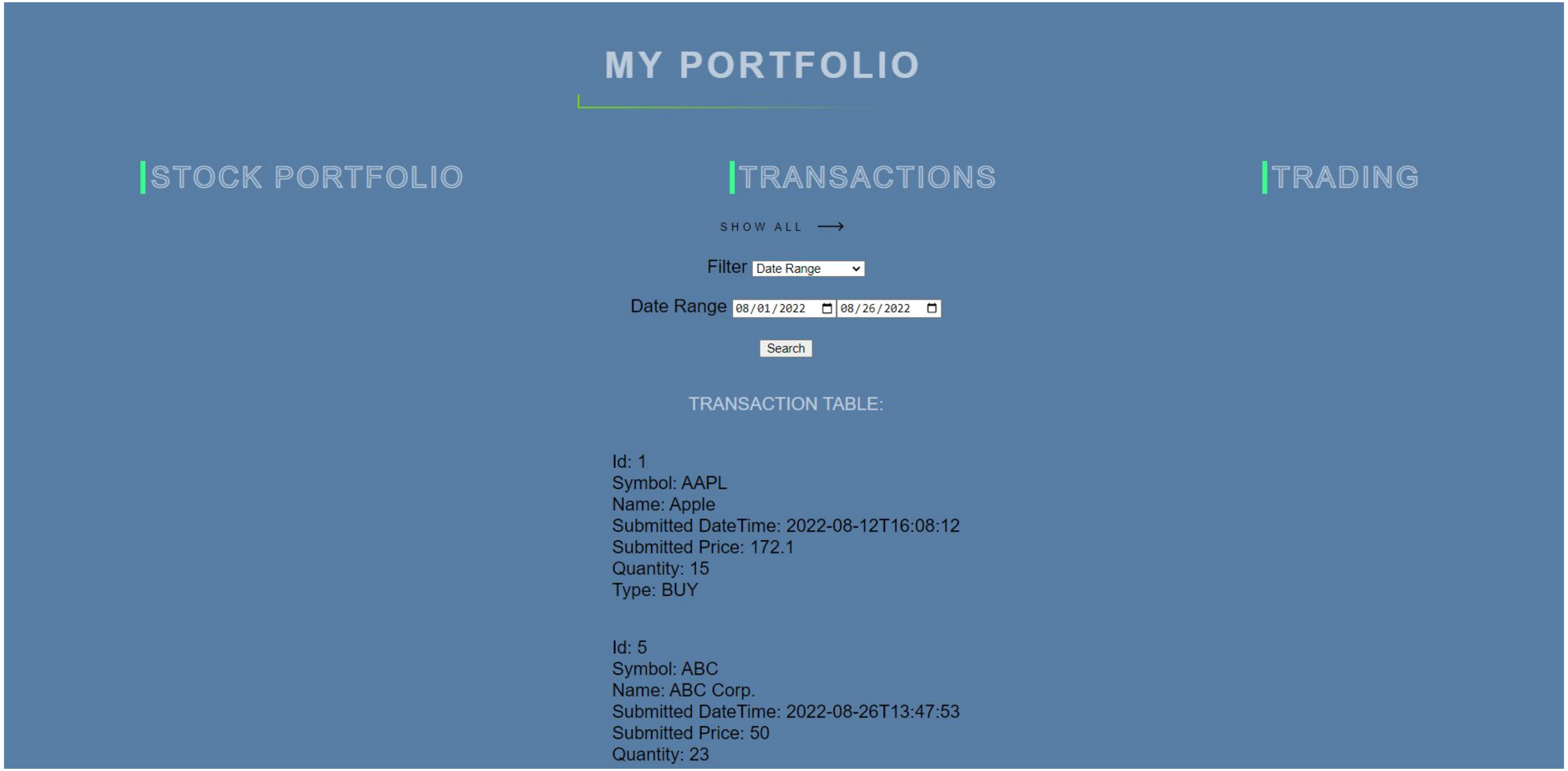
DEF
DEF CORP.
QUANTITY: 50

GHI
GHI CORP.
QUANTITY: 8

MSFT
MICROSOFT
QUANTITY: 10

TSLA
TESLA
QUANTITY: 2

Technical Solution - Transactions Page



MY PORTFOLIO

STOCK PORTFOLIO

TRANSACTIONS

TRADING

SHOW ALL ⟶

Filter: Date Range

Date Range: 08/01/2022 – 08/26/2022

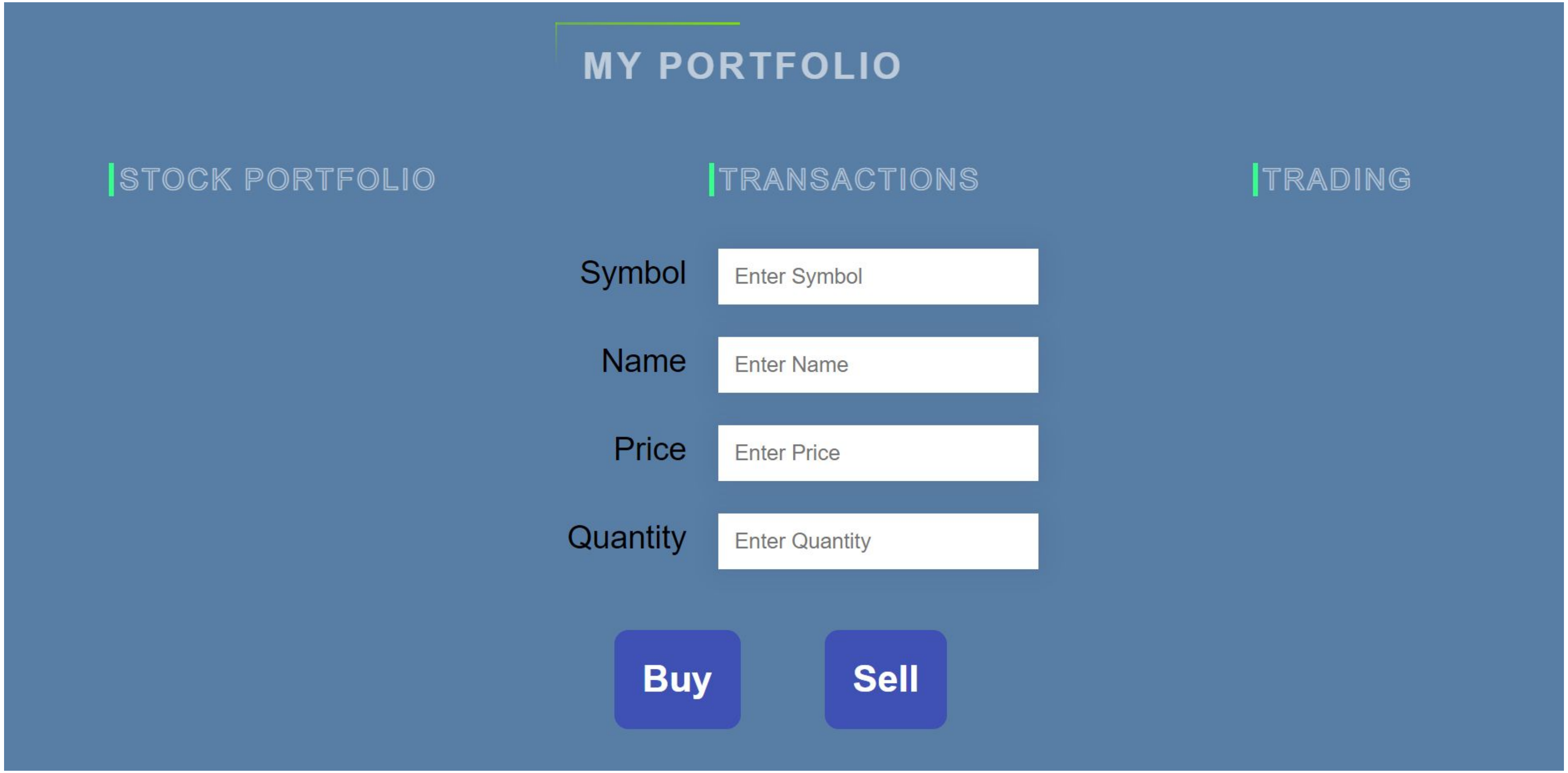
Search

TRANSACTION TABLE:

Id: 1
Symbol: AAPL
Name: Apple
Submitted DateTime: 2022-08-12T16:08:12
Submitted Price: 172.1
Quantity: 15
Type: BUY

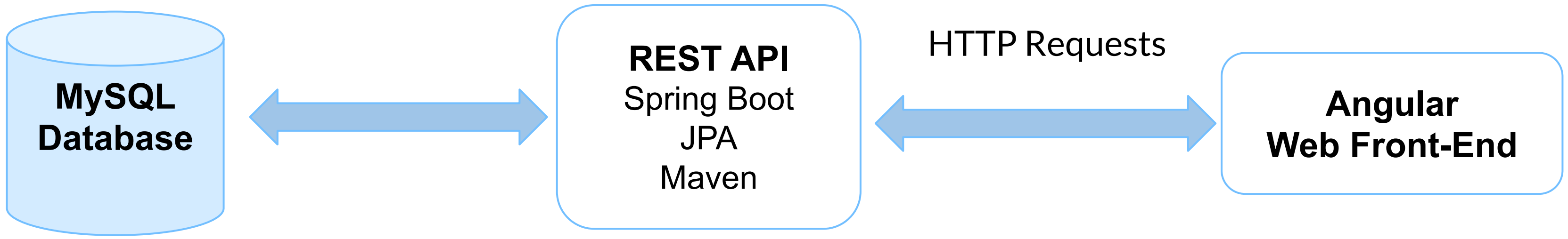
Id: 5
Symbol: ABC
Name: ABC Corp.
Submitted DateTime: 2022-08-26T13:47:53
Submitted Price: 50
Quantity: 23

Technical Solution - Trading Page

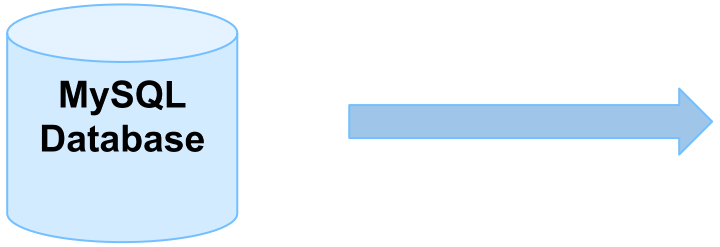


Symbol	Enter Symbol
Name	Enter Name
Price	Enter Price
Quantity	Enter Quantity

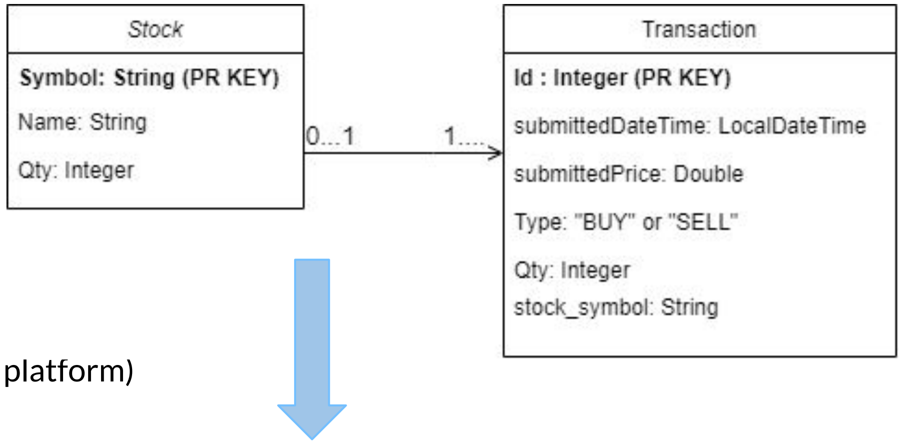
System Architecture: Implementation



System Architecture



Domain Model:



- Used MySQL to create database
- Database is on OpenShift container (cloud platform)

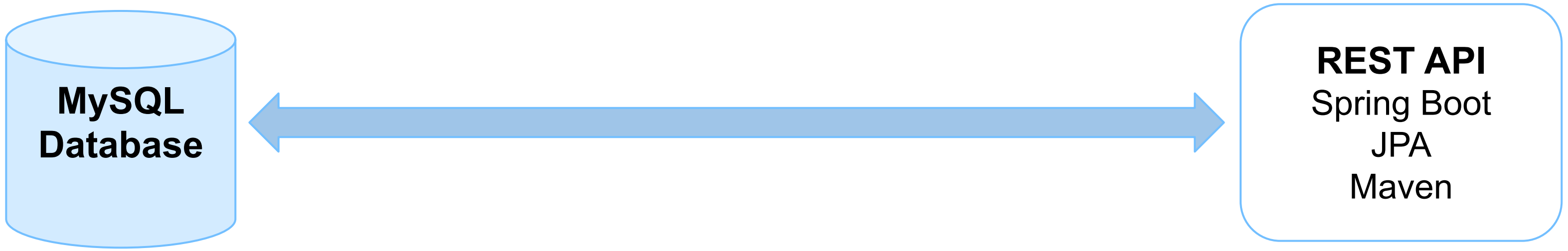
Stock Table

Symbol	Name	Qty
“AAPL”	Apple	10

Transaction Table

ID	Submitted DateTime	Submitted Price	Type	Stock Symbol	Qty
1	2017-01-13T 17:09:42.411	25	“BUY”	“AAPL”	10

System Architecture: Implementation



Functions to retrieve records in Stock table	
Search by	Symbol Name
Update By	Buy Stock Sell Stock

Functions to retrieve records in Transaction table	
Search by	Id Symbol Date range Price range
Update By	Add Transaction

System Architecture

Backend

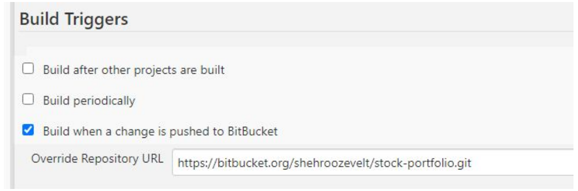
- Application builds are managed by Maven
- Spring Boot is our Project Framework
- JPARepository for CRUD on DAO
- RESTful API used to interact with web services

Frontend

- Build web front-end by Angular : CSS, HTML, TYPESCRIPT

Automated Build and Deployment

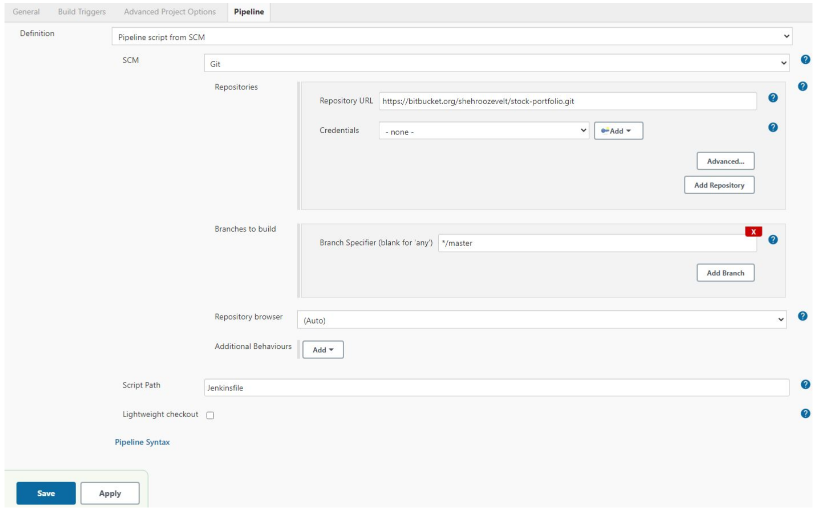
- Jenkins: connect to our BitBucket repository, deploy automatically



Build Triggers

- [ ] Build after other projects are built
- [ ] Build periodically
- [x] Build when a change is pushed to BitBucket

Override Repository URL: https://bitbucket.org/shehroozevelt/stock-portfolio.git



Definition: Pipeline script from SCM

SCM: Git

Repositories
- Repository URL: https://bitbucket.org/shehroozevelt/stock-portfolio.git
- Credentials: - none -

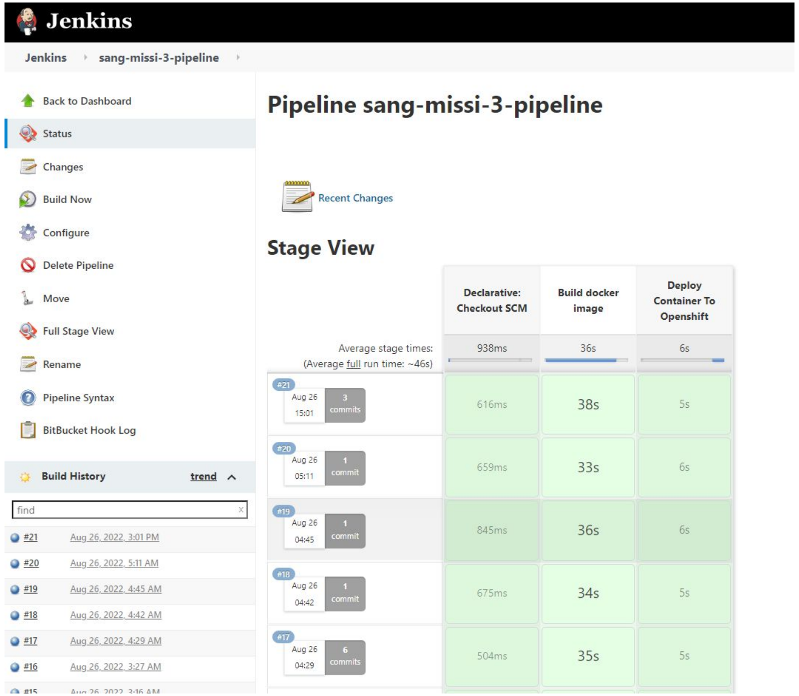
Branches to build
- Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

Additional Behaviours

Script Path: Jenkinsfile

Lightweight checkout: [ ]



Pipeline sang-missi-3-pipeline

Build History trend

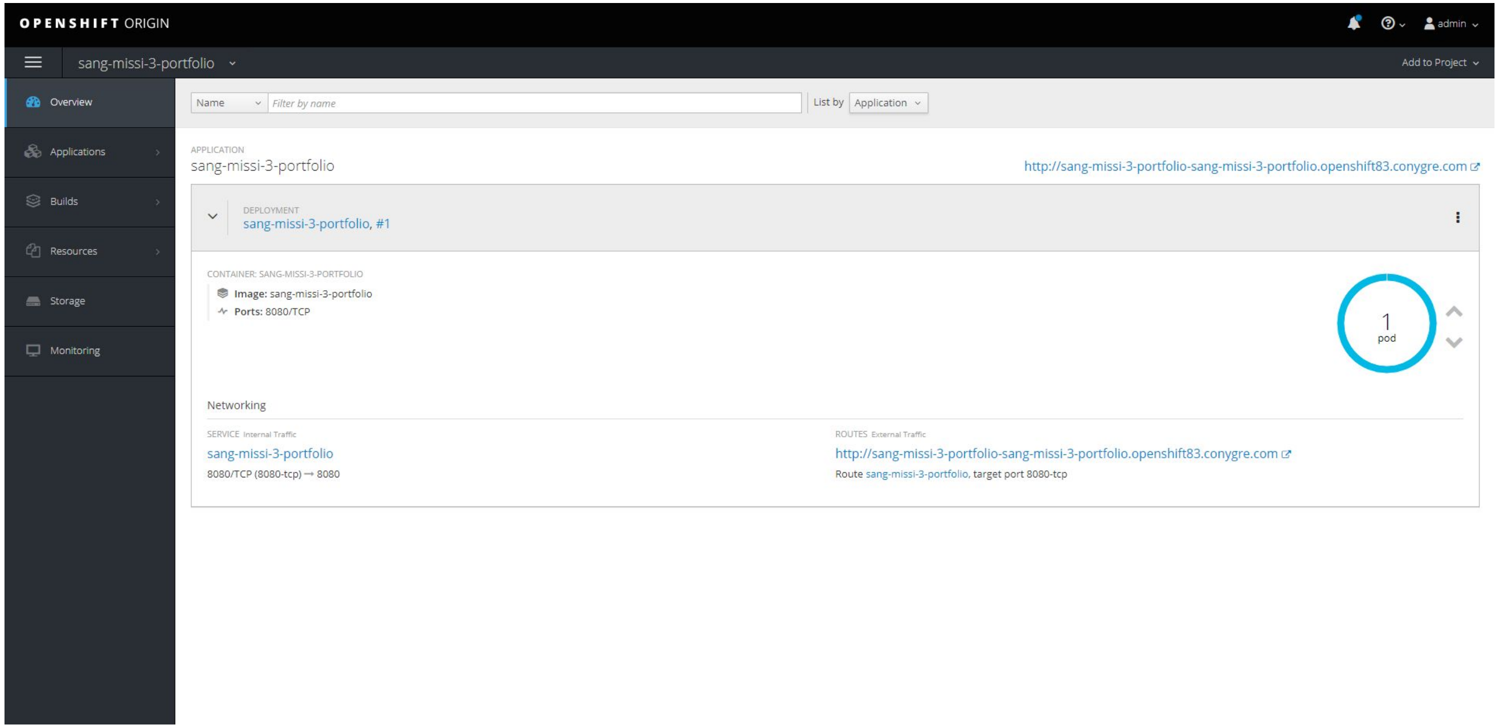
#21	Aug 26, 2022, 3:01 PM
#20	Aug 26, 2022, 5:11 AM
#19	Aug 26, 2022, 4:45 AM
#18	Aug 26, 2022, 4:42 AM
#17	Aug 26, 2022, 4:29 AM
#16	Aug 26, 2022, 3:27 AM
#15	Aug 26, 2022, 3:16 AM

Stage View

	Declarative: Checkout SCM	Build docker image	Deploy Container To Openshift
Average stage times: (Average full run time: ~46s)	938ms	36s	6s
#21 Aug 26 15:01 3 commits	616ms	38s	5s
#20 Aug 26 05:11 1 commit	659ms	33s	6s
#19 Aug 26 04:45 1 commit	845ms	36s	6s
#18 Aug 26 04:42 1 commit	675ms	34s	5s
#17 Aug 26 04:29 6 commits	504ms	35s	5s

Automated Build and Deployment

- OpenShift: holds our database and application on remote server



APPLICATION
sang-missi-3-portfolio
http://sang-missi-3-portfolio-sang-missi-3-portfolio.openshift83.conygre.com

DEPLOYMENT
sang-missi-3-portfolio, #1

CONTAINER: SANG-MISSI-3-PORTFOLIO
Image: sang-missi-3-portfolio
Ports: 8080/TCP

1 pod

Networking

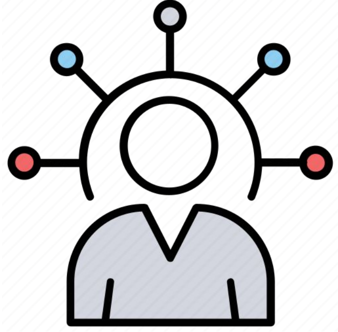
SERVICE Internal Traffic
sang-missi-3-portfolio
8080/TCP (8080-tcp) → 8080

ROUTES External Traffic
http://sang-missi-3-portfolio-sang-missi-3-portfolio.openshift83.conygre.com
Route sang-missi-3-portfolio, target port 8080-tcp

Demo

Challenges

- **Implementing the complete full-stack application**
    - Connecting the Angular with our back-end application with given time
    - Setting up REST API methods for the stock & transaction processes
    - Setting up the application deployment processes
    - Automating the MySQL database setup on Spring Boot
- **Other minor inconveniences include:**
    - Managing code changes across multiple active branches
    - Managing CSS code across the Angular front-end

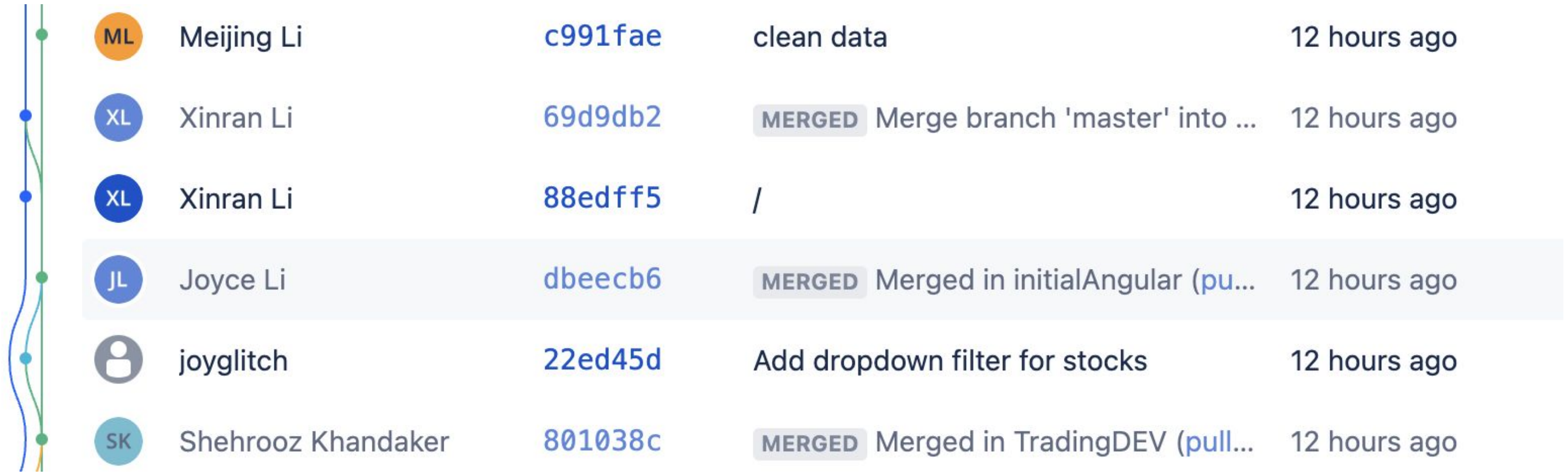


Roadmap for the Future

- Implement additional portfolio services (e.g., net worth, growth/loss over time)
- Connect with external APIs to retrieve real-time stock information(e.g. market price)
- Add charts, tables, graphs to better visualize the user's data
- Extend the portfolio to manage bonds, cash, ETFs, etc.
- Create tests for our application (e.g., JUnit testing)

Collaboration

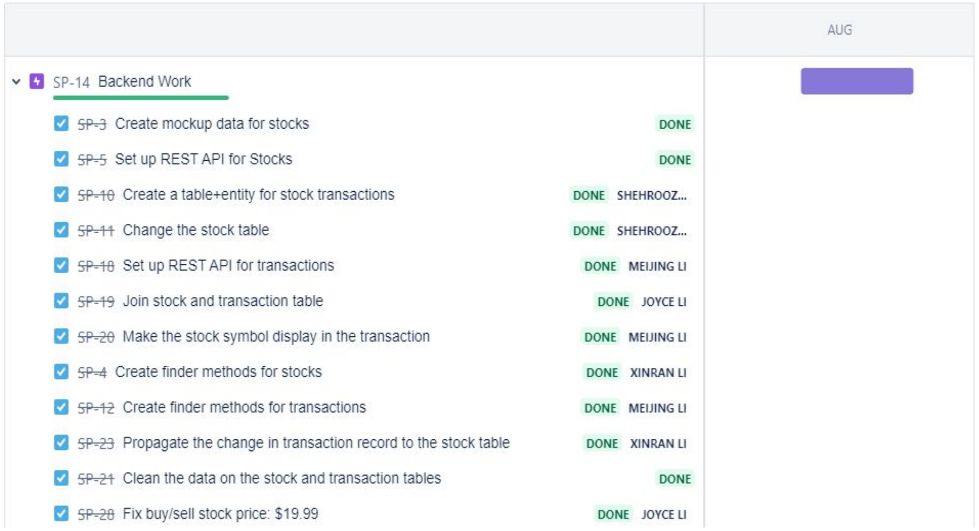
- Bitbucket for version control and working together
  - Work on separate branches
  - Create pull requests and code review before merging branch to master



ML Meijing Li	c991fae	clean data	12 hours ago
XL Xinran Li	69d9db2	MERGED Merge branch 'master' into ...	12 hours ago
XL Xinran Li	88edff5	/	12 hours ago
JL Joyce Li	dbeecb6	MERGED Merged in initialAngular (pu...	12 hours ago
joyglitch	22ed45d	Add dropdown filter for stocks	12 hours ago
SK Shehrooz Khandaker	801038c	MERGED Merged in TradingDEV (pull...	12 hours ago

Collaboration

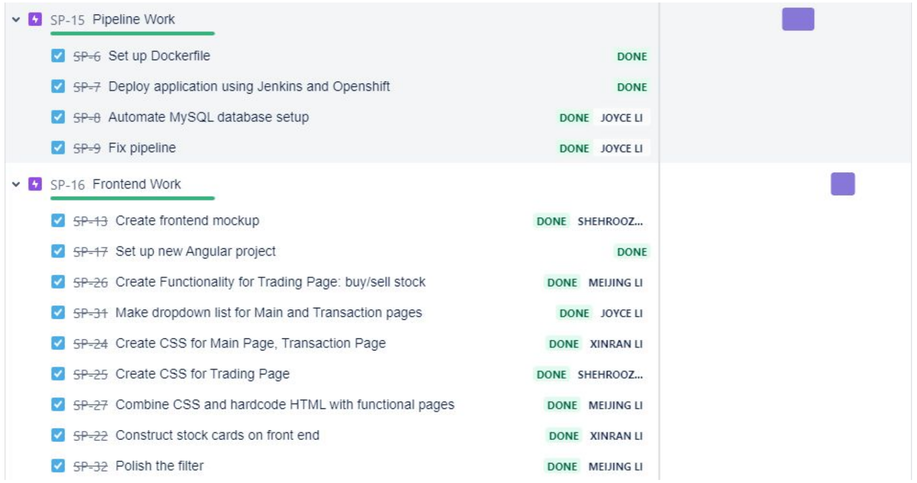
- JIRA for assigning and monitoring tasks



AUG

SP-14 Backend Work

~~SP-3~~ Create mockup data for stocks	DONE	
~~SP-5~~ Set up REST API for Stocks	DONE	
~~SP-10~~ Create a table+entity for stock transactions	DONE	SHEHROOZ...
~~SP-11~~ Change the stock table	DONE	SHEHROOZ...
~~SP-18~~ Set up REST API for transactions	DONE	MEIJING LI
~~SP-19~~ Join stock and transaction table	DONE	JOYCE LI
~~SP-20~~ Make the stock symbol display in the transaction	DONE	MEIJING LI
~~SP-4~~ Create finder methods for stocks	DONE	XINRAN LI
~~SP-12~~ Create finder methods for transactions	DONE	MEIJING LI
~~SP-23~~ Propagate the change in transaction record to the stock table	DONE	XINRAN LI
~~SP-21~~ Clean the data on the stock and transaction tables	DONE	
~~SP-28~~ Fix buy/sell stock price: $19.99	DONE	JOYCE LI



SP-15 Pipeline Work

~~SP-6~~ Set up Dockerfile	DONE	
~~SP-7~~ Deploy application using Jenkins and Openshift	DONE	
~~SP-8~~ Automate MySQL database setup	DONE	JOYCE LI
~~SP-9~~ Fix pipeline	DONE	JOYCE LI

SP-16 Frontend Work

~~SP-13~~ Create frontend mockup	DONE	SHEHROOZ...
~~SP-17~~ Set up new Angular project	DONE	
~~SP-26~~ Create Functionality for Trading Page: buy/sell stock	DONE	MEIJING LI
~~SP-31~~ Make dropdown list for Main and Transaction pages	DONE	JOYCE LI
~~SP-24~~ Create CSS for Main Page, Transaction Page	DONE	XINRAN LI
~~SP-25~~ Create CSS for Trading Page	DONE	SHEHROOZ...
~~SP-27~~ Combine CSS and hardcode HTML with functional pages	DONE	MEIJING LI
~~SP-22~~ Construct stock cards on front end	DONE	XINRAN LI
~~SP-32~~ Polish the filter	DONE	MEIJING LI

Collaboration

- Slack and Zoom for collaboration and communication

Conclusion

- Keep track of user’s stock portfolio and allow purchases of new stocks
- Gained exposure to front-end & back-end technologies

  and how they connect within a full-stack application

Visit Our Repository

https://bitbucket.org/shehroozevelt/stock-portfolio

Thank you for listening!