Добрый день!

Стек примерно такой:

Java Core

- Interfaces, Classes, Modifiers, Synchronized

- Annotations, Collections, Maps, Arrays,

- Reflections, Exceptions, [Multithreading](https://en.wikipedia.org/wiki/Multithreading_(computer_architecture)) Generics

Java 8

- Streams API, Lamdas, Mappers, Optionals, Pipelines, Collectors

Java Persistence

- ORM, JPA, Hibernate, EclipseLink

- Entities, EntityManager, EntityManagerFactory, JPA Repositories

- Persist, Transactional

Spring

- Core, Boot, MVC, Data, Security

- DI, Beans, Services, Qualifiers, Profiling, Components,

- Entities, JpaRepository, CrudRepository

- RouterFunction, Mono, Flux, ServerResponse

- Non-Blocking IO API

RESTful API

- RestEasy, Response, Request

- HTTP, HTTPS, Get, Post, Patch, Put, Delete

- WebSocket, RSocket

- JSON

- Asynchronous request

Security

- Ldap, Oauth, JWT

Servers

- WildFly, Netty, Undertow, Tomcat

Logging

- Log4j2, Sl4j

Maven

- Wrapper, Build, Verify, Deploy, Test, Plugins

Testing

- Junit, Mockito, WebFluxTest, SpringBootTest, WebTestClient

- JMeter

Subversion

- Bitbucket, GitHub, CLI

- Branches, Commits, Merges, Requests

- Jira, Confluence, Slack

UI

- Angular, JSF, TypeScript, CLI

DBMS

- Oracle, PostGres, MS SQL, H2

- SQL, PL/SQL, pgPL/SQL, Pipelines

- Native Query, SqlResultSetMapping

Others

- CI/CD, Docker, Kubernetes, PipeLines, Actions

- Lombok, Swagger, Jackson

- JMeter, Postman

IDE

- Idea, NetBeans, Datagrip, PhpStorm

OS

- Linux MX, Ubuntu, SSH

//==========================================================

Есть большое желание и интерес разработать и внедрить банковскую систему нового поколения, которая придет на замену существующим и устаревшим системам, представленным на белорусском рынке (СофтКлуб (SC-Bank NT), Системные Технологии (ST-Bank), ЛВО), которые используют устаревшие технологии (клиент-сервер, файл сервер), громоздки в разработке и сопровождении).

В настоящее время веду самостоятельную разработку такой системы, на текущий момент мною разработан пилотный проект системы, состоящий из нескольких модулей, готовлю демо-версию для банков, делаю её описание. Исходники выложены на BitBucket и GitHub. Структура БД описана в отдельном документе (power designer), сгенерированы скрипты для создания таблиц БД. Эта система состоит из набора самостоятельных модулей, которые используют следующие инструменты и технологии:

- серверные приложения (модули): Spring Core, Boot, Security, MVC, Data, JPA, Hibernate, Reactive Streams, RSocket, WebSockets, RestFul API, JUnit, Profiling, Stream API

- клиентская часть – Angular JS

- сервер БД – Postgres, MongoDB

Серверные приложения запускаются под линуксом (Java 14, Postgres 13), разработаны с учетом возможной их будущей контейнеризации (Docker, Kubernetes) и параллельной работе при повышенной нагрузке от клиентов. Под каждое приложение разработан набор тестов (JUnit), которые без участия специалиста по тестированию создают и выполняют действия над прикладными сущностями). Эти тесты также выполняюстя при каждой сборке версии проекта (Maven, JUnit), которая происходит при мерже ветки\задания на BitBucket, GitHub.

//==========================================================

Краткое описание подсистемы в отдельном документе, изложенное в виде требований к будущей системе:

//==========================================================

Перечень планируемых модулей в отдельном документе: