

ATELIER II -8h-

COMPETENCES

Concepts :

- Différentes architectures de conception logicielle
 - MVC
 - SOA
 - MicroService
- Avantages/inconvénients des architectures
- Services Web Full Rest
- Gestionnaires de dépendances (basique)

Techno :

- Maven
- Spring boot
- Javascript, AJAX

SUJET

Après analyse de vos premiers prototypes vous décidez de partir sur une architecture **Web Javascript + Web Services Full Rest**. La prochaine étape est de bâtir une plateforme d'achat/vente de cartes. Après inscription des utilisateurs, ces derniers recevront aléatoirement 5 cartes de jeu. Après s'être connectés, les utilisateurs pourront acheter ou vendre des cartes.

Après avoir bien analyser les principes des **Web Services Full Rest**, votre équipe devra assurer la réalisation des fonctionnalités décrites à savoir :

- Création d'utilisateur et connexion
- Achat/vente de cartes sur la plateforme

Votre client souhaite dans un premier temps une **authentification sommaire (ne pas utiliser les modules Security de Spring boot)** et surtout une application fonctionnelle bien **commentée** respectant les standards **FULL REST**.

Développer ces fonctionnalités en respectant les contraintes de votre client en termes de technologie **SpringBoot**. Comprenez l'intérêt de **Maven** et l'usage **d'Entité** pour développer votre application.

Des modifications de la base de données seront nécessaires afin de permettre de stocker les particularités des utilisateurs de l'application.

Après avoir dressé un bilan de réalisation de votre application, réaliser un **tableau comparatif** des architectures logicielles possibles telles que **MVC, SOA, MicroService...**

QUESTIONS

Qu'est ce que des Services Web Full Rest ? Quelles sont les contraintes imposées par ce type de service ?

Qu'est ce qu'un gestionnaire de dépendance ? Maven est-il le seul ? Quel est l'avantage d'utiliser un gestionnaire de dépendances ? Quelles sont les grandes étapes de fonctionnement de Maven ?

Qu'est ce qu'une application classique Entreprise Java Application ? Donner un exemple d'usage avec web service, JSP, JavaBean, EJB et JMS.

Qu'est ce que Spring ? qu'apporte Spring boot vis-à-vis de Spring ?

Qu'est ce que Spring Boot ? Quels sont les points communs/différences entre JEE et Spring Boot ?

Qu'est ce qu'une annotation ? Quels apports présentent les Annotations ?

Comment fait-on pour créer un Web Service Rest avec Spring Boot ?

Qu'est ce qu'un container de Servlet ? Comment fonctionne un container de Servlet ?

Expliquer la philosophie « Convention over Configuration » de Spring boot ?

Expliquer ce qu'il se passe lors de l'exécution «`SpringApplication .run(App.class,args)` »

Qu'est ce qu'un DAO ? En quoi est-ce intéressant d'utiliser ce pattern ? Qu'est ce qu'un Singleton ? Que permet de réaliser les Entity dans Spring boot ? Est-ce spécifique à SpringBoot ?

Combien d'instances avez-vous créées lors de l'usage de «Service » en Spring boot? Pourquoi ?

Que fournit le CRUD Repository de Spring boot ? Que sont les CRUD ?

Qui réalise l'implémentation de la méthode `findByProperty` lors de la création d'un repository en Spring Boot ?

Comment gère-t-on les relations One To One, One to Many et Many to Many avec JPA ?

Qu'est ce qu'une Architecture SOA ? Qu'est ce qu'une architecture Micro Service ? Il y a-t-il d'autres architectures existantes ? Quels sont leurs avantages/inconvénients ?

Axe de travaux indépendants :

Dépendance / Maven

Dépendance / Injection

Spring et Spring boot

Les Entreprise Java Application Standard

REFERENCES

***(*indispensable*) **(*important*) *(*optionnel*)

[Spring boot complete course]

- *** <https://javabrainsthinkific.com/courses/take/springboot-quickstart/multimedia/2775600-introduction>
 - Introduction spring Boot (ts les chapitres) 1h15
 - Spring MVC: The View Tier (ts les chapitres) 45min
 - Booting Spring Boot (Using application properties) 6min
 - Spring Data JPA : the Data Tier 1h
 - Deployment and monitoring [optional]

[Spring boot outline]

- ** <http://www.springboottutorial.com/introduction-to-spring-boot-for-beginners>

[Spring Boot] fast start :

- ** <http://www.mkyong.com/spring-boot/spring-boot-ajax-example/>
- * <https://spring.io/guides/gs/rest-service/>
- * <https://spring.io/guides/gs/accessing-data-rest/>

[Spring boot complete example]

- * <http://www.springboottutorial.com/creating-rest-service-with-spring-boot>
- *** <http://websystique.com/spring-boot/spring-boot-rest-api-example/>

[Dependency injection]

- *** <https://www.youtube.com/watch?v=Hm2fTv9wikI>

[Spring Boot JPA]

- *** <https://www.callicoder.com/hibernate-spring-boot-jpa-one-to-many-mapping-example/>

[Maven]

- ** <https://maven.apache.org/what-is-maven.html>
- ** <https://javapapers.com/jee/maven-in-10-minutes/>
- ** https://www.youtube.com/watch?v=XulPrVct_xQ

[Architecture logicielle]

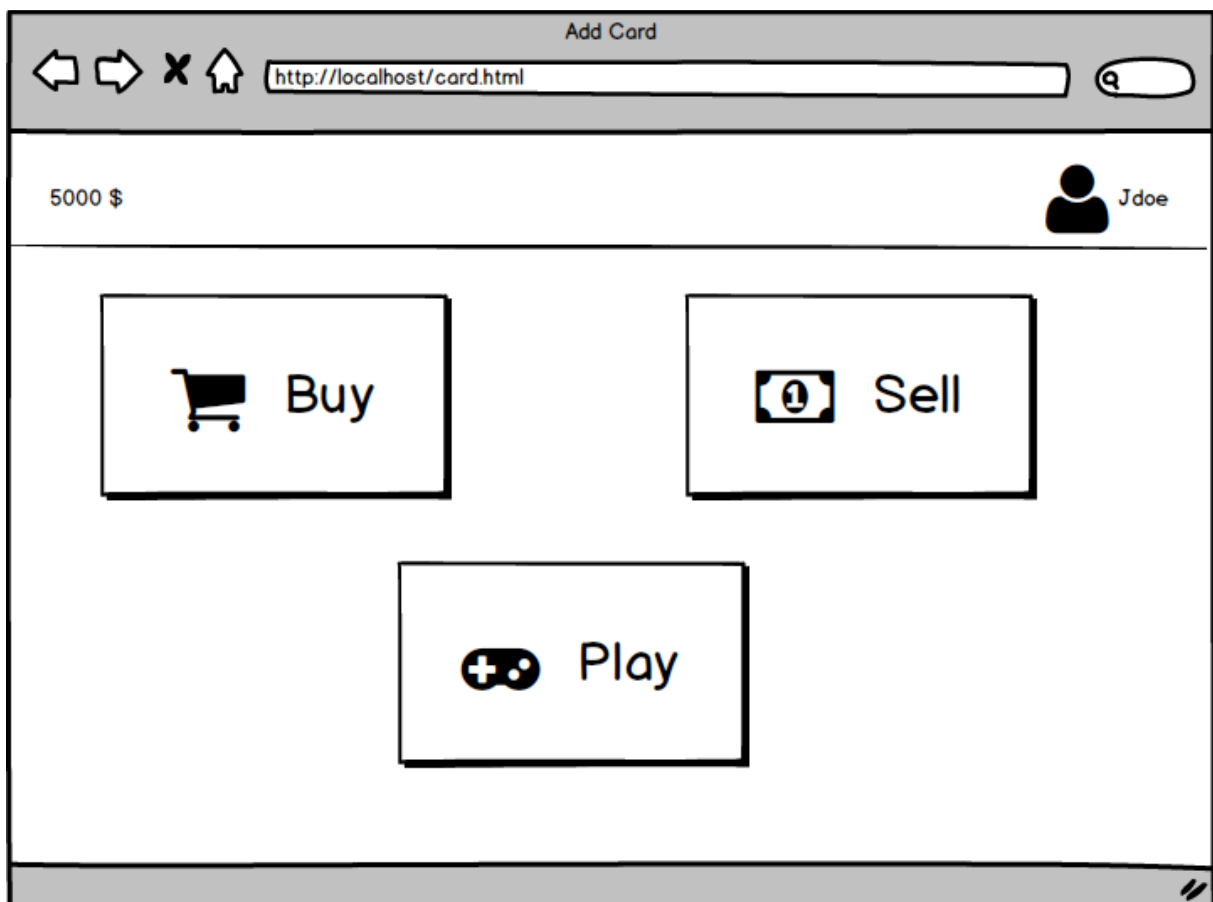
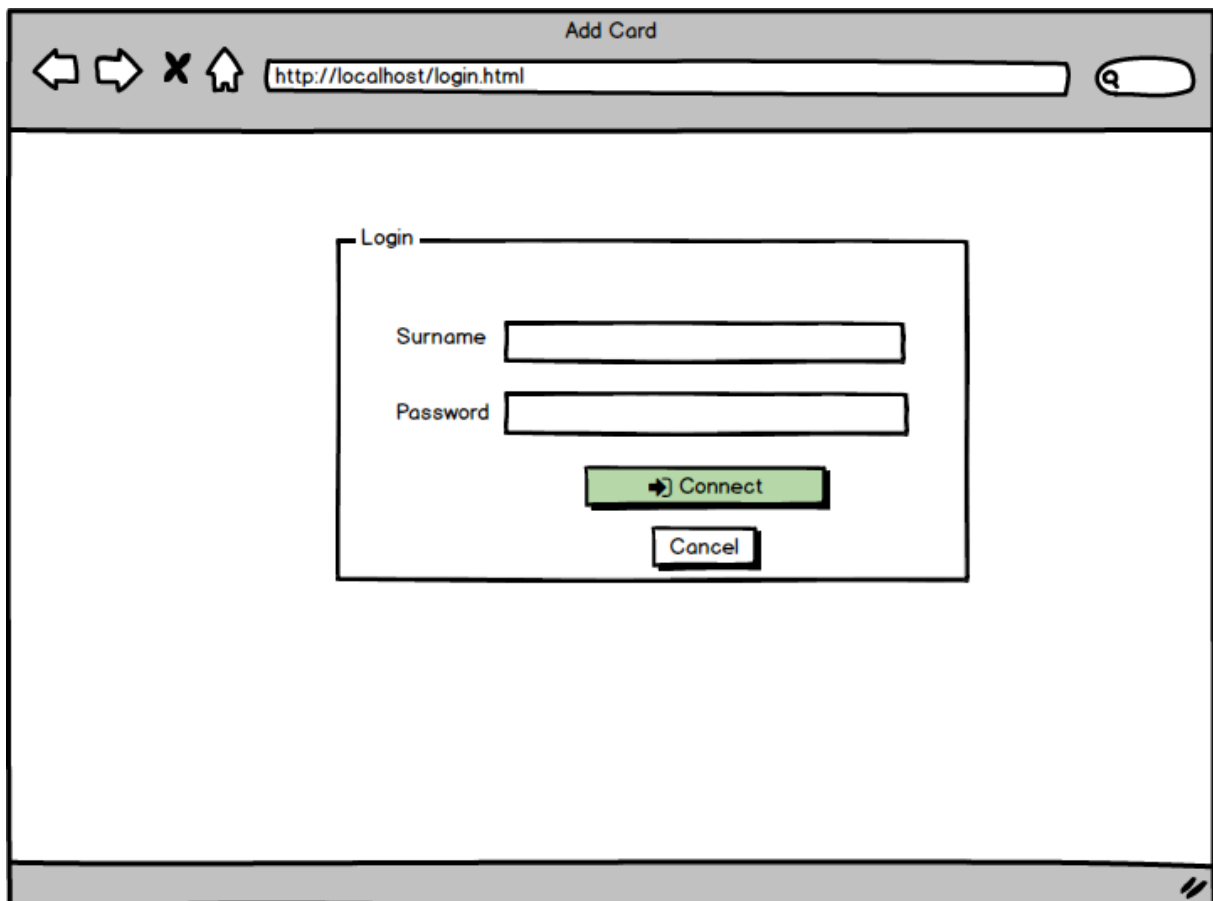
- ** <https://www.openshift.com/promotions/microservices.html>
- ** <http://www.nakov.com/blog/2011/06/29/software-architectures-client-server-multi-tier-mvc-mvp-mvvm-ioc-di-soa-cloud-computing/>

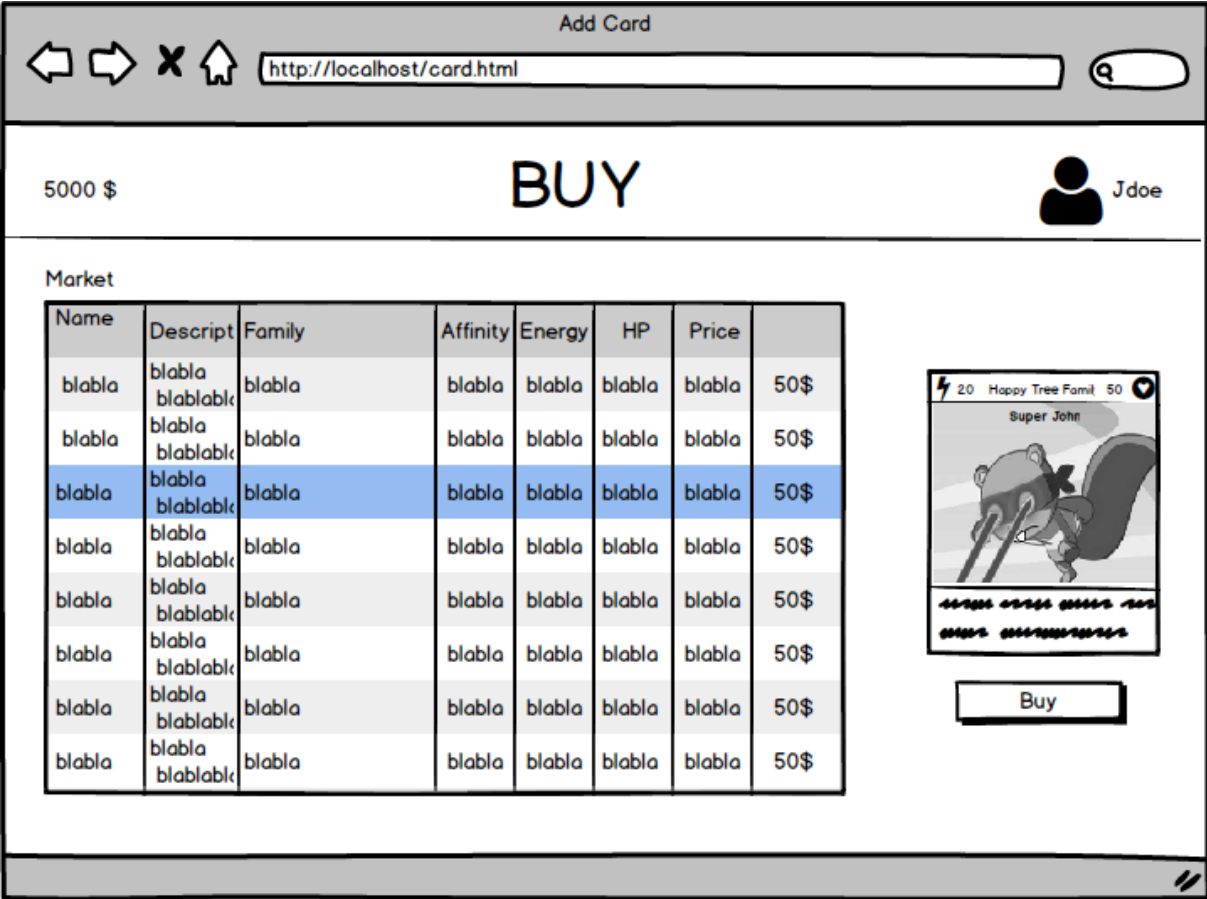
ELEMENTS DONNES

- IDE configuré
- 1 sketch
- Une base de données

- 1 exemple simple de webservice + Spring boot
- Visuel Html UI Semantic + base javascript d'interactions

The diagram illustrates a web browser window titled "Add Card". The address bar contains the URL "http://localhost/addUserhtml". The main content area displays a form titled "UserForm". This form includes four text input fields labeled "Name", "Surname", "Password", and "Re-Password". Below these fields are two buttons: "Cancel" and "+ OK". The entire interface is rendered in a hand-drawn, sketch-like style.





← → ✕ 🏠


Add Card

http://localhost/card.html

🔍

5000 \$

SELL


Jdoe

My CARD

Name	Descript	Family	Affinity	Energy	HP	Price	
blabla	blabla blablabla	blabla	blabla	blabla	blabla	blabla	50\$
blabla	blabla blablabla	blabla	blabla	blabla	blabla	blabla	50\$
blabla	blabla blablabla	blabla	blabla	blabla	blabla	blabla	50\$
blabla	blabla blablabla	blabla	blabla	blabla	blabla	blabla	50\$
blabla	blabla blablabla	blabla	blabla	blabla	blabla	blabla	50\$
blabla	blabla blablabla	blabla	blabla	blabla	blabla	blabla	50\$
blabla	blabla blablabla	blabla	blabla	blabla	blabla	blabla	50\$
blabla	blabla blablabla	blabla	blabla	blabla	blabla	blabla	50\$

⚡ 20 Happy Tree Famil 50 🔒

Super John



=====

=====

SELL