Projet Conception, Projet Synthèse

*NAuction*

Par Nathaelle Fournier et Quoc Huan Tran

420-C61-IN PROJET SYNTHÈSE

Présenté à Jean-Christophe Demers

Technique de l’informatique

Cégep du Vieux Montréal

2 mars 2023

Table des matières

[Maquettes 3](#_Toc127454720)

[Conception UML 4](#_Toc127454721)

[Structure de données externes 6](#_Toc127454722)

[Contraintes & Éléments de Conception 10](#_Toc127454723)

[Interface utilisateur 10](#_Toc127454724)

[Structure de données 10](#_Toc127454725)

[Patrons de conception 11](#_Toc127454726)

[Expressions régulières 12](#_Toc127454727)

[Algorithme 12](#_Toc127454728)

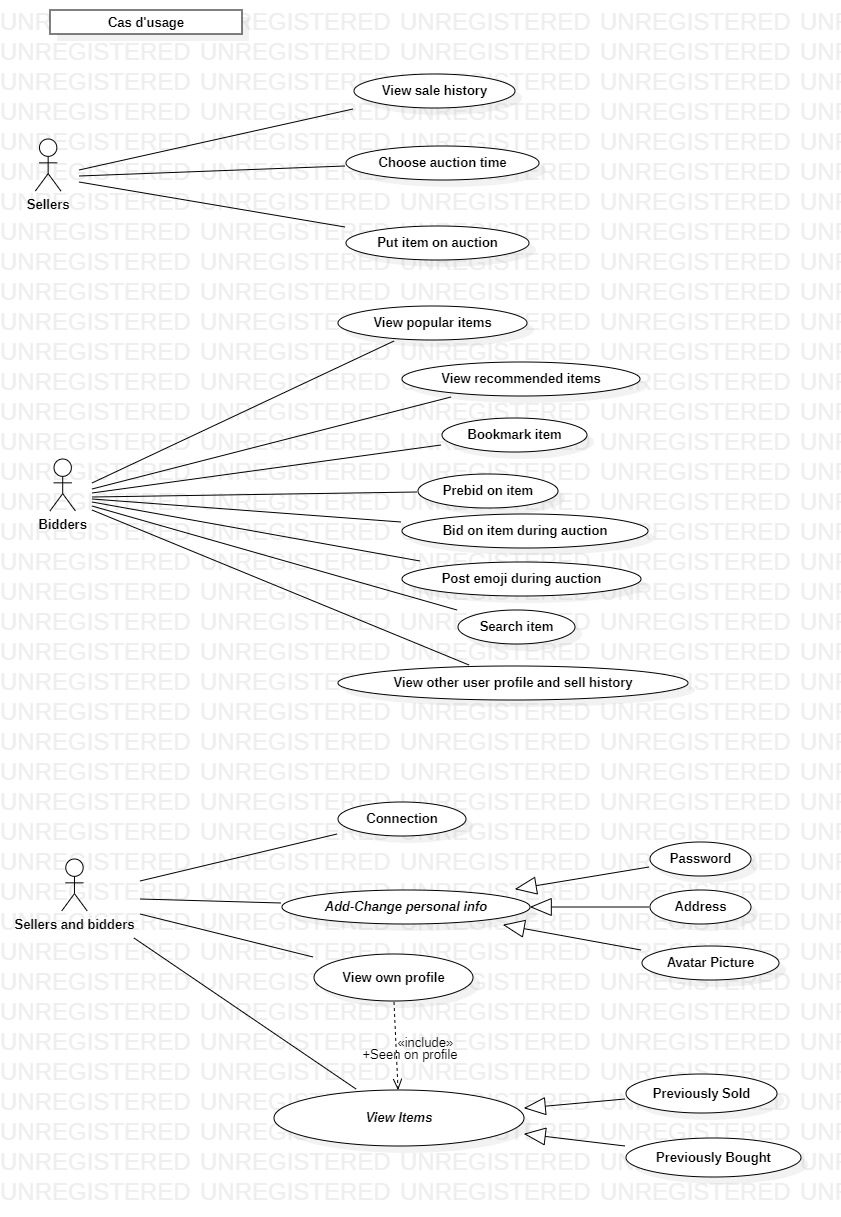
[Mathématique 12](#_Toc127454729)

[Veille Technologique 12](#_Toc127454730)

# Maquettes



# Conception UML



# Structure de données externes

Nous utiliserons une base de données basée sur PostgreSQL. Voici la première ébauche de code pour celle-ci.

CREATE TYPE "status" AS ENUM (

'waiting',

'in\_auction',

'sold'

);

CREATE TYPE "tag" AS ENUM (

'Antique',

'Art & Sculture',

'Automobile',

'Bijoux & Accessoire',

'Collection',

'Livre & Manuscrit',

'Meuble',

'Monnaie',

'Musique',

'Sport',

'Vaisselle & Coutellerie',

'Vêtement'

);

CREATE TABLE "users" (

"id" INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

"username" varchar(32) UNIQUE NOT NULL,

"name" varchar(32),

"lastname" varchar(32),

"email" varchar(128) UNIQUE NOT NULL,

"password" varchar(132) NOT NULL,

"dateofbirth" timestamp NOT NULL

);

CREATE TABLE "address" (

"id" INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

"id\_user" int NOT NULL,

"street" varchar(128) NOT NULL,

"apt" int DEFAULT null,

"city" varchar(32) NOT NULL,

"country" VARCHAR(32) NOT NULL,

"postal\_code" CHAR(6) NOT NULL

);

CREATE TABLE "items" (

"id" INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

"name" VARCHAR(32) NOT NULL,

"description" VARCHAR(256) DEFAULT '',

"current\_status" status NOT NULL,

"bid\_count" int DEFAULT 0,

"min\_increase" numeric(5,2) DEFAULT 1,

"id\_seller" int NOT NULL,

"auction\_on" timestamp UNIQUE NOT NULL,

"room" int

);

CREATE TABLE "bought\_items" (

"id\_user" INT NOT NULL,

"id\_seller" INT NOT NULL,

"id\_item" INT UNIQUE NOT NULL,

"bought\_on" timestamp

);

CREATE TABLE "tag\_list" (

"id\_item" INT,

"id\_tag" tag

);

CREATE TABLE "bids" (

"id\_item" int NOT NULL,

"id\_user" int NOT NULL,

"amount" numeric(,2) NOT NULL,

"submited\_on" timestamp NOT NULL

);

COMMENT ON COLUMN "items"."room" IS 'auction\_on & room must be unique';

COMMENT ON COLUMN "bids"."amount" IS 'must be higher then precious price with the same id\_item';

ALTER TABLE "address" ADD FOREIGN KEY ("id\_user") REFERENCES "users" ("id");

ALTER TABLE "tag\_list" ADD FOREIGN KEY ("id\_item") REFERENCES "items" ("id");

ALTER TABLE "bids" ADD FOREIGN KEY ("id\_item") REFERENCES "items" ("id");

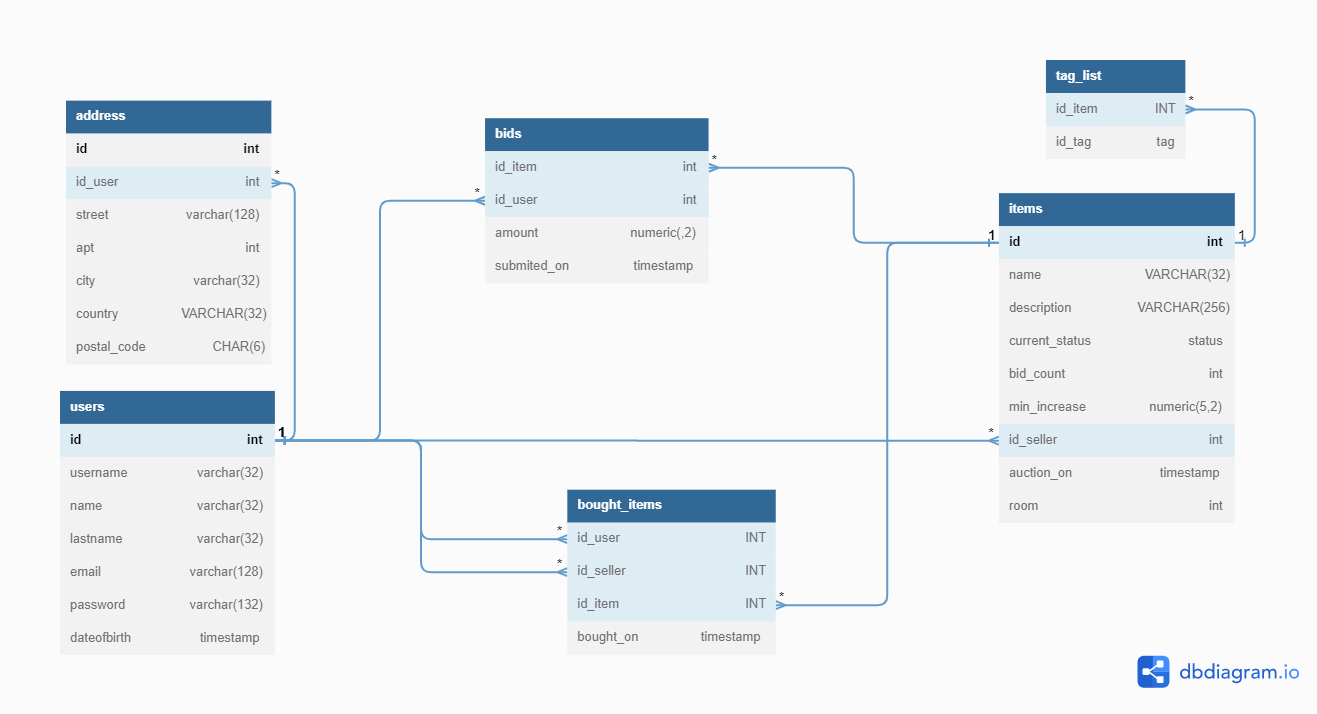
ALTER TABLE "bids" ADD FOREIGN KEY ("id\_user") REFERENCES "users" ("id");

ALTER TABLE "items" ADD FOREIGN KEY ("id\_seller") REFERENCES "users" ("id");

ALTER TABLE "bought\_items" ADD FOREIGN KEY ("id\_seller") REFERENCES "users" ("id");

ALTER TABLE "bought\_items" ADD FOREIGN KEY ("id\_user") REFERENCES "users" ("id");

ALTER TABLE "bought\_items" ADD FOREIGN KEY ("id\_item") REFERENCES "items" ("id");



# Contraintes & Éléments de Conception

## Interface utilisateur

## Structure de données

Dictionnaire pour les images qui sont associé à un item en particulier ?

Liste chainée pour les placements dans le chat que l’on retrouve dans un encan

Tuple pour un pseudo d’usager avec le prix qu’il a annoncé pour un item dans un encan.

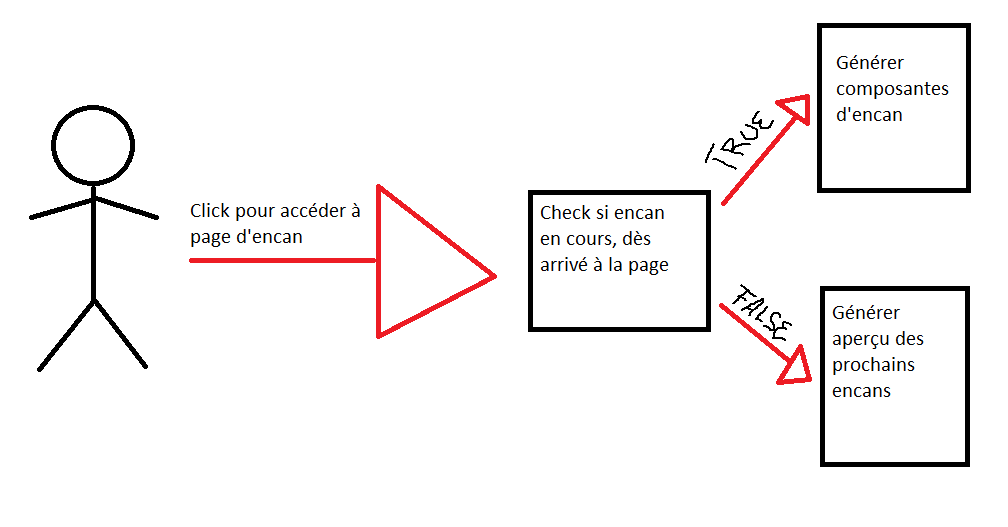
## Patrons de conception

Les patrons de conception que nous avons choisi de spécifiquement implémenter dans notre projet sont *Observer*, *Iterator* et *State*.

Nous avons choisi *Observer*, car il s’agit d’un patron évident pour tous projets web qui demande quelconques interactions de l’utilisateur. Il sera implémenté dans la majorité des cas pour appeler une nouvelle composante *React* à l’écran lorsque l’on appuie sur le bouton approprié.

Pour l’itérateur, nous pensions l’intégrer dans la liste chainée que nous pensions concevoir. Nous pensons que celui-ci sera utile pour parcours la liste au cours d’un encan et nous avons aussi décider de l’implémenter pour avoir un exemple complet de l’intégration d’une liste chainée dans un projet.

Nous pensons utiliser le patron du *State* pour la page où l’on retrouverait les encans. Puisque ceux-ci ne seraient qu’accessible que lorsqu’il y en a un cours. Advenant le cas où il n’y en aurait pas, seul un horaire de ceux à venir serait afficher. Ainsi, dépendamment de l’état de la page, nous allons générer différentes composantes *React*.



## Expressions régulières

Nous pensons utiliser les expressions pour la recherche d’article en particulier. Sois l’usager entre quelque chose dans la barre de recherche et on retourne une liste des articles dont le nom correspond aux données reçues.

## Algorithme

Algorithme de recommandation qui se tient sur les tags des items visités, ainsi que ceux acheté précédemment par cet usager.

## Mathématique

Pour calculer un minimum de mise automatique se basant sur le prix courant de l’item

## Veille Technologique

*React* pour la réutilisation de composantes graphiques, ainsi que pouvoir les modifier à tous moments grâce au *Virtual DOM*.