

Name of the project : « Drop Da Bomb »

Team members : VINCENT Pierre, DEUTSCH Rémi, LOIGNON Lucas

## Table of contents

Introduction .....	2
Project goals : .....	3
Steps of the projet .....	4
Analysis of needs and conception of the database .....	4
Creation of the database .....	4
Creating a starting set of data.....	4
Setup of web server with “Tomcat web container” .....	4
Analysis of needs of treatments server-side and implementation (Servlets).....	5
Development of a user web interface (HTML / CSS / JSP) .....	5
Development of administration web interface (HTML / CSS / JSP) .....	5
Bonus : Development of an user mobile (Android) interface (JAVA Android / XML) .....	6
Independent / additional / interdisciplinary tasks.....	6
Summary of langages and technologies used : .....	6
Annexes.....	7

## Introduction

Game principle : *Drop Da Bomb* is a board/card game designed for smartphone (Android systems).

The game is based on a collection of Cards (**Game Cards**). A player (**Account**) try to gather all, or at least the best, of them, in order to create an efficient gaming set of cards (**Deck**).

Thoses cards could be obtained from several ways but the main way is through the shop (**Shop**), which allow players to buy with a specific money some sets of cards available (**Packs**). There is several kind of card, with differeents levels of rarity.

A game (**Match**) oppose two player in an specific arena with specific rules and gaming modes. Alternately, both players can use one card from their Deck in order to try to make the objective (« the bomb ») moving in the opponent's game zone.

If the bomb explode or is present in player's camp when the time period expires, the game is lost for this player ; the other one win the game and could be granted with rewards.

Our team aim to develop all the features described before, excepted the Match itself. We our going to use a lot of modern and popular technologies and programming langages, mostly new for us at the moment.

The purpose is to design and create a fully and fonctionnally database, to install and configure a web server equiped with the Tomcat technologie in order to perform most of the actions & treatments, to design and code an web interface pour the user (the player), and if it's possible an web interface for administration purposes and even an Android interface for the user.

This project means a lot for us as it is a long time and well reflected idea ; we are trying to take advantage of this « school » discipline with the objectif of setting up the structure of the game project, as robust and upgrading-ready as possible.

The concrete realisation of the game represent such a big load in terms of work, skills and time that we may not try of sucess it entirely, but we will try to make, in this discipline, as much work as we can so that we could come to the end of this project in our extra-time, some day.

All the technologies used will be described after the presentation of the steps of the project.

### Project goals :

- Set up a consistent, efficient, complete and expansion-open database for the game and all of its components (including future possible needs)
- Set up a communication system Server/Client using HTTP requests and JSON files.
- Design one (several if possible) interface (for the client only at the moment, the administration one optionally) in order to perform all kind of actions (such as account creation, connexion, modification, and some gaming actions like display and manipulation of inventory and decks)

## Steps of the projet

### Analysis of needs and conception of the database

*Finding and defining all the data linked to account managing, inventory managing, decks managing, social relations, matchmaking, shop, etc...*

*Anticipating as possible the final version of the application (context far away of this « school project »).*

*Keeping a close watch at the consistency, non-redundancy and effectiveness of tables and relations of our database.*

### Creation of the database

*From the Conceptual Model of Data done before, writting scripts of creation for different tables of the project.*

*Choosing types and ranges (possibles values allowed) of each attribute, promoting the integrity and upgrading possibilities of the database. Once the database has been created, a minimum of changes may be done on it.*

### Creating a starting set of data

*Writting insertion scripts for some specific tables, in order to be able to start tests and requests on dependent tables.*

*Exemple : some tables, like the Cards one, does not depend of the user action (like the Account one for exemple) ; values of thoses tables are almost fully known at the moment by the conceptors (us).*

### Setup of web server with "Tomcat web container"

*In the aim of manipulating and handling data from database, from web or android user interface, we made the choice of the Tomcat technology : this provides us a way of creating programs (written with JAVA EE) called « Servlets » that can handle and perform some specific actions requested by the client. These sevlets are the perfect intermediary between client requests and results bases on the database content. See [Figure 1](#) from Annexe for a representation of the communication schema.*

*So, in a first time, we have to install un web server (Apache) on our machine (that we will call « the server » in this document). In a second time, we install above it un « web container » (Tomcat) which allow us, later, to call servlets from a specifig address on the configurated port.*

## Analysis of needs of treatments server-side and implementation (Servlets)

*As the Tomcat server is ready, the goal is now to determine, design and develop several servlets in order to answer to all of the client's requests.*

*From simple servlets like the account connexion one or the account creation , to most complex servlets such as the « pack opening » from the Shop, a lot of Servlets has to be designed, respecting as possible the security and the integrity of server-side data. Furthermore, thoses Servlets has to be fully expansion-opened.*

## Development of a user web interface (HTML / CSS / JSP)

*As the database and servlets are now done, the next step is to be able to create, modify or manipulate data by a client (a player).:*

*So we are going to design a web interface, accessible from the web browser that provide a way to perform thoses features :*

- *Connexion to an account*
- *Creation of an account*
- *Updating account informations (email, password, pseudo, etc..)*
- *Display of the Inventory (cards, icons, skins, ..)*
- *Display and modification of a Deck*
- *Display and purchase of Shop's items*

## Development of administration web interface (HTML / CSS / JSP)

*The goal is to create a new interface similar to the previous one but this time designed to answer administration needs. The access will be reserved for administrators of the game and will provide for them a way to :*

- *Modify, blocking, banning or deleting player's account*
- *Creating and adding an item (card, icon, skin, ..)*
- *Updating an existing item*
- *Creating a pack for the shop using existing items*

*This interface will allow administrators to moderate player's accounts, modify, updating or adding in-game content (to the database, actually). The interface has to be clear, simple and will provide a handy and safe way to interact with database.*

Bonus : Development of an user mobile (Android) interface (JAVA Android / XML)

*Purposes are the same as those of the « [Development of a user web interface](#) » part, excepting this time the client is a mobile-one (Android)*

Independent / additional / interdisciplinary tasks

- **Algorithmic** : design algorithms to perform specific actions, such as the automatic random cards choosing of Shop's packs, according to defined rules.
- **Design** : conception of several gaming items, mainly cards, icons and skins ; official logo & theme of the application (Photoshop)
- **Global Design** : find a global theme for our project/game (colors, fonts, backgrounds..) and set up a beautiful design, modern, efficient and as user-friendly as possible for our user's interfaces..

Summary of languages and technologies used :

- Conception : UML, MCD, MLD
- Communication : AJAX, JSON, XML
- Development : JAVA EE, JAVA ANDROID, JSP, MySQL, HTML, CSS
- Frameworks : MaterializeCSS
- Other : Photoshop
- Teamworking through **GIT** and **Discord** (software for oral/text chat for teams).

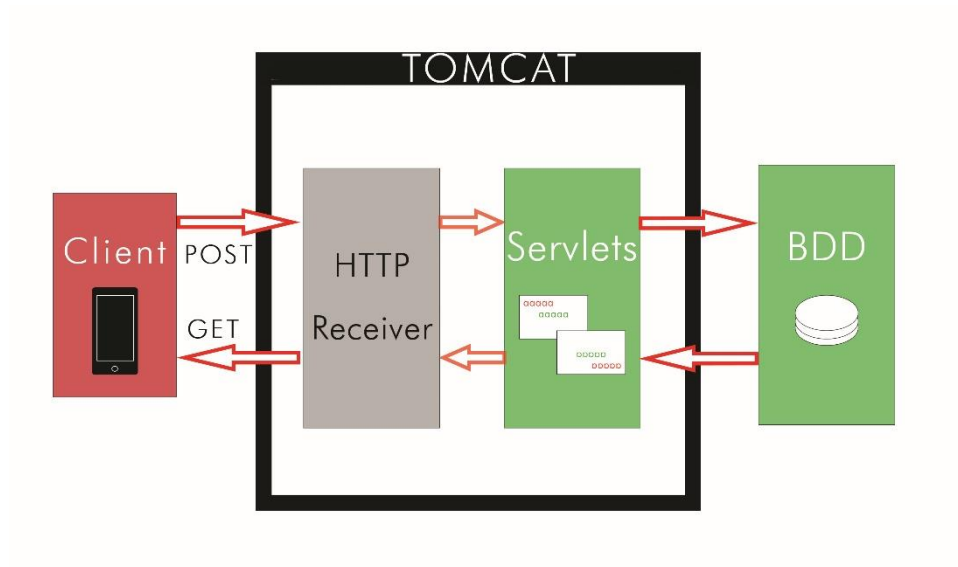


Figure 1 – Structure of our systems communication setup

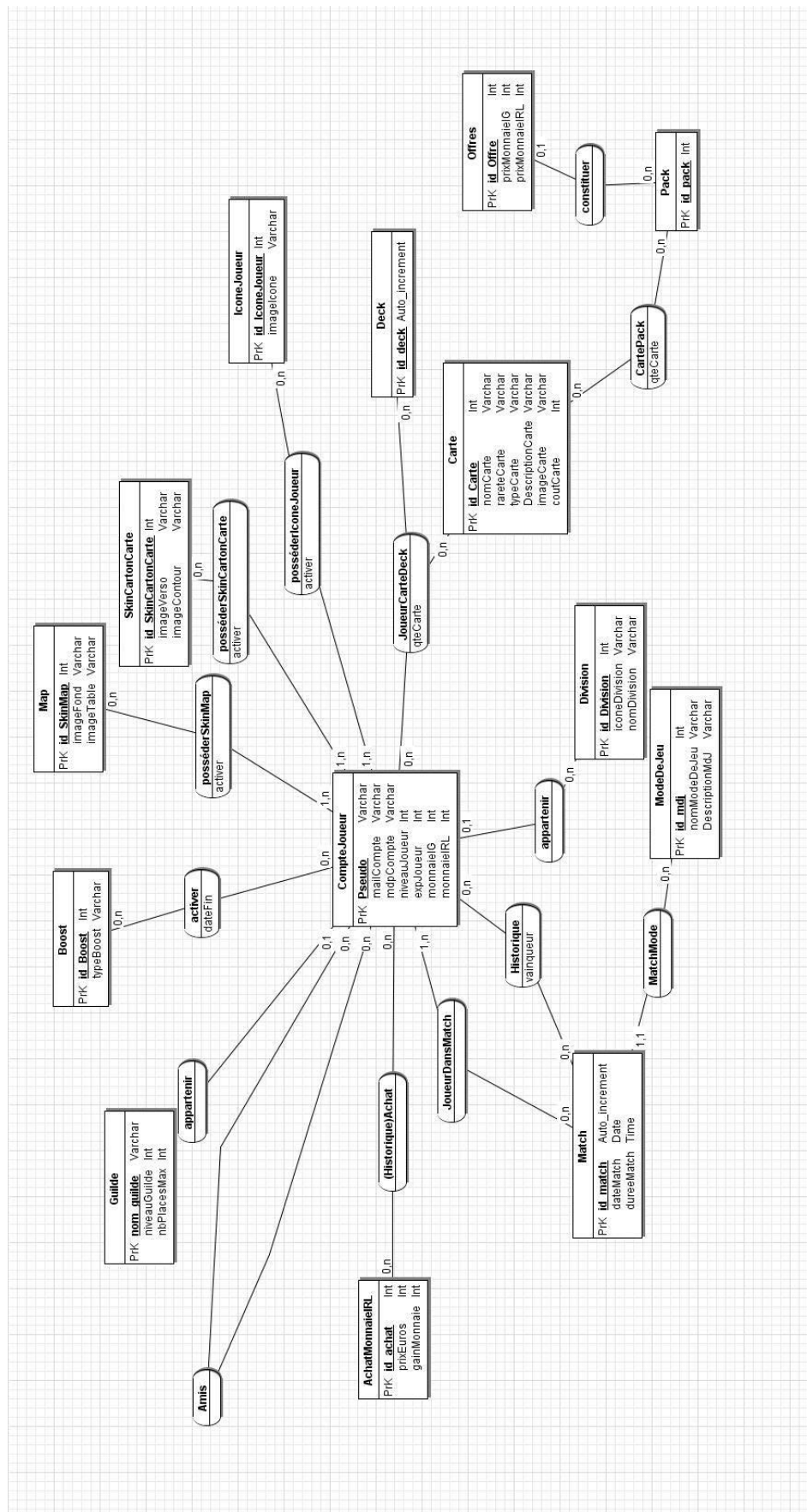


Figure 2 – First representation of the database (new version incoming)