Teamwork 2025/26 (v1.0)

1. Introduction (same as Sprint 1)

The objective of the teamwork is to develop an information system for collectors. The system must be developed for the Web (e.g. **website** or webportal) and all information about the collections must be stored persistently in a relational database. The project occurs in two **sprints** or iterations.

2. Description

2.1 Context (same as Sprint 1)

The intended Information System (IS) is to manage collections (automotive miniatures, locomotive miniatures, stamps, coins, comic book classics, trading cards, among others) of collectible objects/parts/items, hereinafter simply called **items**. The information system to be developed will allow efficiently the CRUD (**C**reate-**R**ead-**U**pdate-**D**elete) of all information regarding the collections.

2.2 Global needs (same as Sprint 1)

The system must allow managing more than one type of collection. When creating collections, the characteristics of the collection must be defined (name, date of acquisition, etc). It must be possible to see the items in a specific collection and also the events where a specific collection occurred or will occur.

The system must allow classifying each collectible item according to parameters considered relevant, such as importance (value between 0 and 10), monetary value, weight (in grams) or others considered relevant. The system must also allow ordering the items according to the criteria defined above. If an item is part of more than one collection, its information will not be repeated.

Based on the example in **Figure 1**, we can see a website of itsms from various types of collection. By selecting the StarWars miniatures collection, it is possible to view the various items that make up that specific collection.

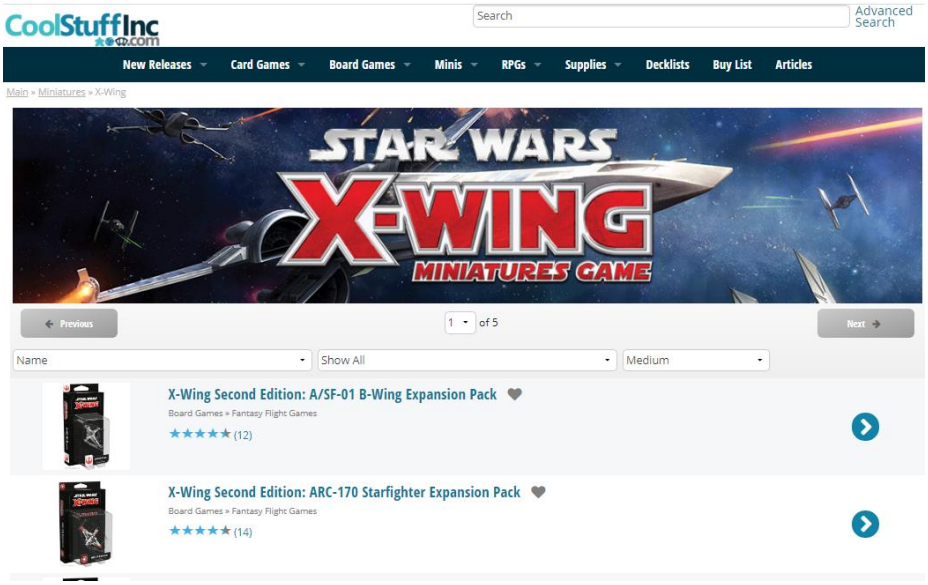

Figure 1: Example of a collection (coolstuffinc.com).

After choosing a particular item, a page will open (**Figure 2**) with all the information about the specific item of the StarWars miniatures collection.

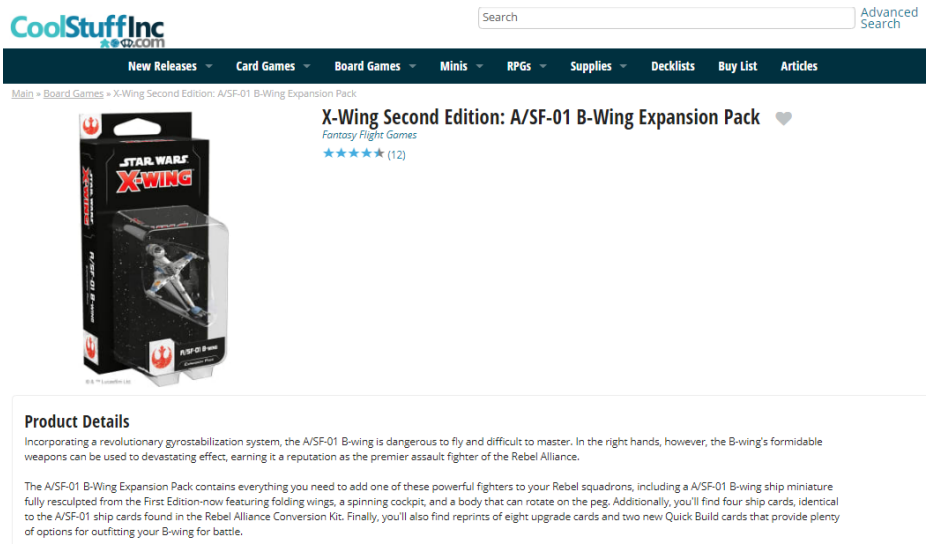

Figure 2: Web page of a specific item of a collection.

It must be possible to list all the items or just the item that make up a given collection. Previews only show part of the item information (**Figure 1**). For a complete description of a given item, the information system (IS) must refer to a specific page of the item (**Figure 2**).

For a better idea of your needs, consult collections or collectors' websites. For example: https://www.coolstuffinc.com, or https://www.revistaqui.com.br.

Just to clarify, the objective of this IS **is not** an online shop. The objective is to create an IS so that users can manage its own collections.

2.3 Minimal system

For the first sprint (**Sprint 2**), which corresponds to the second phase of the work, the system must include at least:

- A mysql database, that allows you to store information about collections, items, events, associated users (collectors) among other relevant data that each team have considered in Sprint 1.
- Make the web portal created in the first phase of this work (sprint 1) dynamic, meaning that it will be possible to view, insert, modify, and delete (CRUD) all information contained in the database;
- It is mandatory to use PHP language in server-side communication;
- A user will only have access to manage their collections and related events through a single sign-in system:
    - It should also be possible to register new users and/or change their information through the user profile page;
    - all users (registered or unregistered) can view the collections of all users.
    - Access to create, update, and delete personal data is restricted to registered users, and they can only manage their own data (i.e. only the creator of the data has permission to modify or delete it).
- The addition of features that prove to be important for the IS will be valued.
    - As example, loading part of the information (users, collections and items) from a .csv file to quickly enrich the IS it is sometimes very useful. It is also useful to be possible to export that same information into a .csv format.

3. Considerations about the information (same as Sprint 1)

Whenever there is ambiguity in gathering the problem's requirements, the team can take on new considerations (new attributes, relationships, tables, classes, etc.), as long as they make sense in the context of the problem. For basic information (collections, items, events) consider at least the following attributes:

- Collection
    - name
    - type
    - creation date
    - …
- Item
    - name
    - Importance
    - weight
    - price
    - Data of acquisition
    - …
- Events
    - name
    - localization
    - …
- User
    - Name
    - Date of birth
    - Email
    - …
- …

4. Development Process and Technologies

Teams are made up of **4/5 members**. The development process must be iterative.

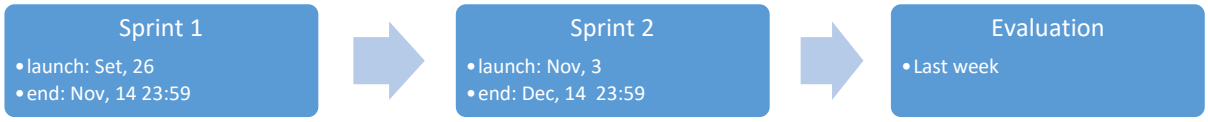

Figure 3: Development process.

Process for Sprint 2:

1. In the initial phase, stabilization of the ER diagram of the conceptual model;
2. Transition from the conceptual model to the logical model;
3. Implementation of the physical model: creation of the database and some relevant data that will support the proposed project;
4. Next, division of the team by functionalities/pages in order to:
    a. Create a set of classes/methods in PHP that support CRUD associated with users, events, collections, items, etc;

Remains mandatory to use the GIT tool and the GitHub repository. The repository will be the same one used in Sprint 1.

Frequent commits remain mandatory, and the use of technologies from sprint 1 continues to be part of the Information System. The products indicated in sprint 1, namely the ER model and database implementation script, must be added to the repository.

The database must be implemented with MySQL and the SGBD will be the phpMyAdmin (other can be used). The team will need to develop classes, methods, and program logic using the PHP programming language.

The system must ensure the separation between the various business and interface layers and data (i.e. DAL class – e.g. https://www.geeksforgeeks.org/dbms/data-access-layer/). Use HTTP methods according to the REST style as indicated in the classes course unit. Interaction with this Information System must be done through the interface created in sprint 1. However, this interface may be modified to better meet the needs encountered (The team can modify Sprint 1 interface to meet new requirements).

Additional guidance, or new requirements, may be given during the project.

5. Evaluation

At the end, the student must demonstrate the ability to develop an information system, focused on the Web.

5.1 Sprint 1 (30% weight + 2.5% report)

Submission in Moodle until Nov, 14th at 11:59 pm (Moodle time).
Submission in a file (. zip) with

1. The release of the interface developed in PHP, HTML/CSS/Javascript/ (NetBeans project or standalone PHP, HTML, CSS and javascript files) for the desired system;

2. Script for creating the database and its tables with relevant data (inserts into the tables that make up the database);
3. Report containing (a template of this document is available in moodle):
    a. team composition;
    b. individual task description (who made what);
    c. work limitations (what was not accomplished and why);
    d. self and hetero evaluation (a template of this document is available in moodle);
    e. other specifications and/or considerations that the team finds relevant.

The evaluation may take into account the progress of the work. Autonomy, depth of work, quality of models, code, development process, as well as effective functionalities are factors to be taken into consideration.

5.2 Sprint 2 and final presentation (Weight of 40% + 2.5% report + 15% presentation)

Submission in Moodle until Dec 14th at 11:59 pm (Moodle time).

The presentation is mandatory, with a weight of 15%. The presentation will be given by the team, and will include a demonstration of the application, and the team or individuals may be asked questions about the technical details of the application.

The evaluation may take into account the progress of the work. Autonomy, depth of work, quality of models, code, development process, as well as effective functionalities are factors to be considered.

The time for each presentation will be according to the scheduling file, which will be made available during the week of submission.

6. Version control

V1.0	Initial version