DEVELOPMENT OF THE TRIVIAL GAME

Proposal of the problem

The Game Company *NoGame* is going to start a new product line based on question/answer games that should work in several platforms. As a first step, they want to build a variant of the Trivial Pursuit game [1], although the goal is that other similar games based on question/answers could be created in the future.

In the first stage, they need to obtain a large number of questions to feed the different games. To that end, they want to build a simple application that reads files that contain the questions/answers, processes those files reporting any errors, and stores them in a database.

The files with the questions/answers will come from existing test banks. One of those test banks uses the GIFT format [2]. The application will use an internal representation of the questions using JSON format which will facilitate its storage in the database. The Company is planning to use MongoDb [3], although this decision has not yet been taken.

It is expected that the application can be executed in two phases. The first phase parses the files with the questions and generates the internal representation. The second phase takes the questions in that internal representation and stores them in the database. The execution of these 2 phases will be controlled by a company operator who could decide when he wants to execute each phase and even could automate that execution to be done at some time intervals.

It is important to warrant that the conversion is correct, easing the load process debugging. To that end, the solution that is being sought doesn't require an interactive interface but it requires several input options (select name of file to load, input format, name of output file, output format, etc.) and should enable to check that the internal representation of the conversion does not produce errors.

As it is planned to automate the conversion process to be executed at some specific times, the company doesn't require the application to be very time efficient or that the conversion process can be done interactively.

Although in this first step, the company only requires the conversion from GIFT format, the company is considering that the solution should allow other formats in the future. For example, the company is considering to process also questions in XML format like QTI [3].

Identification of the stakeholders

Trivial Development team:

They are the ones in charge of creating the trivial game and develop the architecture in order to facilitate the creation of future trivia games.

Some of their objectives are:

- Develop the Trivial game in an efficient way, controlling the costs and following the standards of usability thinking in the final user.
- Create a reusable architecture that can be implemented in other games based on question/answer mechanisms.

People in charge of NoGame:

These are the directors of the corporation, in charge of the budget, from which they allocate funds for the project and administrating the different games developed.

Some of their objectives are:

- The duration of the project should be short and the costs as minimum as possible.
- Get the maximum Profit.
- Get an application whose architecture can be used in other variants of the game.

Development team of NoGame:

They are the ones in charge of developing future trivia game that will use the architecture developed by Trivial Development team as a help in the process.

Some of their objectives are:

- Use that architecture to facilitate the development of new versions of the game.
- Finding the best technological alternatives to implement the game, and communicating them to the Trivial Development team.

Players of the Game:

The final users of the product, they want to play the game in an easy and intuitive way.

Some of their objectives are:

- Playing the game in an easy way without having to spend too much time trying to understand how the application works.

Identification of the quality attributes

Availability

- The system must have available all the functionality 24-7(every day at any time).

Modifiability

- Scalability of the system, we probably may do changes in the program.
- Facility to change the parser system, the questions and answers files format.
- Facility to change the connection with the database if in the future we want to change the database administrator program.

Performance

- We have to make sure that our systems responds to the user in a reasonable amount of time.
- Latency should be as low as possible in order to provide the user a good experience within the application.
- Throughput (number of events that take place within a given amount of time) should be as high as possible.

Security

- We have to ensure the security and strength of the system.

- We have to provide a good architecture in order to avoid non-ethical attack against our system that try to break into the server or that try to mislead the system.
- The system should be able to avoid DDoS attacks (very important in an application like this one).

Usability

- The system must be the most intuitive as possible, in order to help the user to understand its functioning.
- A manual must be available for any user that doesn't understand how to use the system.

Adaptability

- The system must work for any operating system.

Time to market

- Short development cycle

Cost-Benefit

- Null development cost

First approach to the solution

First approach

Once we have analyzed the requirements of the system, we have arrived to a possible solution based on a Batch system. That is a way of reducing the interation with the user avoiding an extra cost in specialized operators of the system focused on managing the translation and the databe. This solution will be formed of an application that will process text files with different formats extracting its content and storing it in a database.

Risks related to the solution

1. Guarantee of the database integrity in case of erroneous data introduce.
2. Protection against the introduction of not valid data both the parser and the database.

To avoid the problems caused by the risks mentioned, the following solutions are proposed:

1. Creating a backup of the database.
2. Creating a series of preconditions and exceptions that controls the flow of data that enters the system.

Quality attributes and stakeholders

Stakeholders vs Attributes	ST-01	ST-02	ST-03
AT001		X	
AT002	X		X
AT003	X		X
AT004			X
AT005		X	X
AT006		X	X
AT007		X	X
AT008	X	X	X
AT009	X	X	X
AT010	X	X	X
AT011		X	X
AT012			X
AT013			X
AT014	X	X	
AT015	X	X	

Business description of the solution

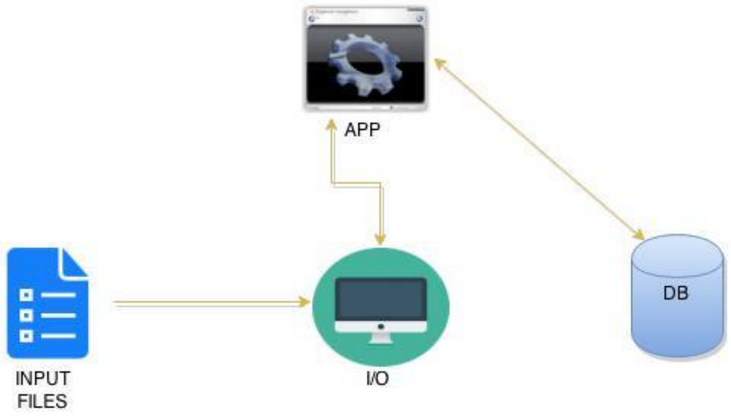
In our application we will have 2 differentiated systems

The Parser:

This section of the system will be the one in charge of reading and processing the input files with the questions and send it to the database.
It will have the responsability of taking care that the format of file is respected and will rise and exception it case it contains non fitting information.

The Database:

Will store the information formated by the parser until the future games will request it.



Quality scenarios

Scenario Number	Source of the stimulus	Stimulus	Environment	Artifact	Response	Response Measure	Affected quality attribute
1	Process the file questions	Data Processing	Explotation	Parser	Effective process time	Effective time > 90%	AT001
2	Modify the way the question is processed	Change the questions processing system.	Explotation	Parser	Sustitute the algorithm without altering other parts of the system	Other parts of the system altered = 0	AT002
3	Change the lexical analizer	Sustitute the lexical analyzer of the questions	Development	Parser	Sustitute the parser without altering other parts of the system	Parts of the system altered = 0	AT003
4	Change the target database	Introduce a new database system	Communication with the database.Development	Database system	Sustitute the database without altering other parts of the system	Parts appart from the database altered = 0	AT004
5	Acess to the Database processed data	Taking a question/answer from the database	Communication with the database	Database system	The answer/question is given from the database	Amount of data losed < 1%	AT005
6	Acess to the Database processed data	Perform the communication among the parser, the I/O and the database	Communication with the database	Database system	The resquested data is given to the other parts of the application	Request time< 5 s	AT006
7	Illigal access to the database	A user request data from de database without permissions	Communication with the database	Database system	Restricton is raised to avoid this access	Successfull illegal accesses < 1%	AT007
8	Acess to the raw documents	A user request data from de question/answer storage with no permissions	Illegal Explotation	Questions storage system	Restricton is raised to avoid this access	Successfull illegal accesses < 1%	AT008
9	Excess of access solicitude	A big amount of users request data from the data base.	Communication with the database	Database system	Find the origin of the massive access and lock it	Maximum number of allowed access attempts per user = 10	AT009
10	Introducing "help" in command line	Wrting the string "help" in the command line	Explotation	Command line	The help is shown in the command line	Availability = 100%	AT010
11	Changes in the system setup of the development team.	The operating system is changed.	Development	Development systems and teams	Sustitute the operating system without altering other parts of the system	Capabilities lost after the operating system changed = 0%	AT011
12	System in development	End of the development	Development	Development systems and teams	Low development time	Development time < 3 weeks	AT012
13	System in development	Cost of the development	Development	Development systems and teams	Low development cost	Total cost=0	AT013