Trivial_i1a Documentation: Final Deliverable

Mario Germánico Pascual Tomás
Daniel Gonzalez Gómez
Edgar Herrero Uría
Adrián Prada Gil
Ainhoa Longo Pérez

Development of the trivial game

Proposal of the problem **3**
Identification of the stakeholders **3**
Trivial Development team 3
People in charge of NoGame 4
Development team of NoGame 4
Players of the game 4
Identification of the quality attributes **4**
Availability 4
Modifiability 4
Performance 4
Security 5
Usability 5
Adaptability 5
Time to market 5
Cost-Benefit 5
First approach to the solution **5**
First approach 5
Risks related to the solution 5
Quality attributes and stakeholders **6**
Business description of the solution **6**
Graphical user interface 6
Logic 6
Database 6
API REST 6
Quality scenarios **7**
Component diagram **8**
Summary 8
Details 10
DB 10
Children 11
DataBase 10
Relationships 12
QuestionManagerInterface 11
AccauntManagerInterface 12
Relationships 13
PLAY FRAMEWORK 13
Children 14
LOGIC 13
Children 15
GUI WEB 14
Children 15
Relationships 16
GameLogic 15
Children 16
Ports 17
Resident Components 17
QuestionManagement 16
Resident Components 18
Views 17
QuestionSelector 18
Relationships 19
QuestionDownload 19
Relationships 21

QuestionArray....21
Relationships....22
API REST....22
Children....24
MovementManager....23
Relationships....24
N/A....25
Relationships....27
AcountManagement....26
Children....27
Resident Components....28
API controller....27
UserLogin....27
Relationships....29
API REST Interface....29
ScoreManager....29
Relationships....31
UserAdministrator....32
Relationships....33
User's system manual....34
General functionality....34
GIFT format....34

DEVELOPMENT OF THE TRIVIAL GAME

Proposal of the problem

This time, what we want is to make the Trivial game online. That is, it should support multiuser and let the users store their punctuation. The architecture must be scalable and offer good answer times to the actions of the connected users.

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 web browser.

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 Web application. This is because we need the application to manage several players playing at the same time. Due to that the game interface will be implemented to work in a web browser, but logic and database are expected to have a minimum amount of changes compared to the desktop version.

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 3 differentiated systems

Graphical user interface:

This section of the system will be the one in charge of showing the game board and the questions, making possible to interact with the Trivial Game. To do this, Html and CSS will be used. Events will be programmed using JavaScript and the navigability will be implemented with Scala.

Logic:

This part of the application performs all the operations related to the functionality of the game, it mainly get the questions, manage the players, the position of the player along the track…

Database:

Composed of text documents: one for the questions and another one for the users and all the information related to the login and the punctuation.

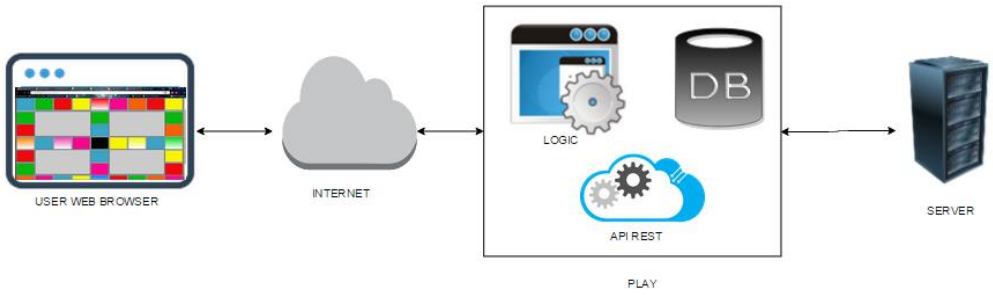
API REST:

REST is a software architectural style for distributed hypermedia systems. It is used to describe any simple web interface that uses XML and HTTP, without the additional abstractions of the message interchange pattern-based protocol.

We used the API REST for the maintenance of users. It is used to obtain information about the users and showing it.

But we could also have used it for obtaining the questions of the trivial and showing them during the game.

Scheme representation of the problem:

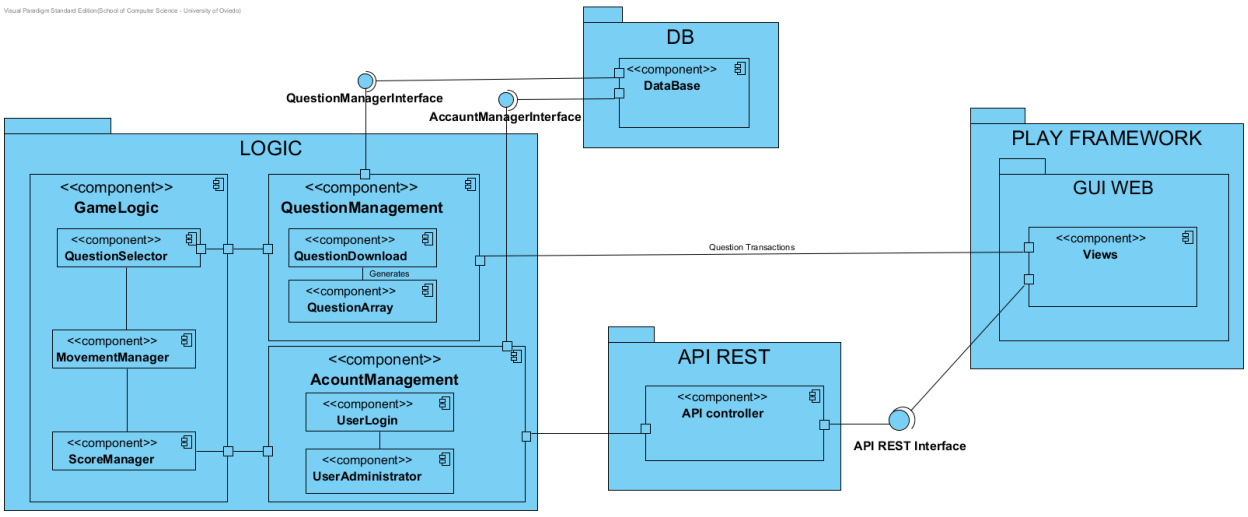


Quality scenarios

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

Component Diagram

Trivial_i1a-ThirdDeliverable



Name	Value
Name	Trivial_i1a-ThirdDeliverable
Author	Trivial_i1a
Create Date Time	07-may-2015 9:25:23
Last Modified	10-may-2015 21:24:58
Teamwork Create Date Time	0
Show Component Attributes	false
Show Component Operations	false

Summary

Name	Description

DB	Database package
DataBase	Database component
QuestionManagerInterface	
AccauntManagerInterface	
PLAY FRAMEWORK	Play environment
LOGIC	Logic package including all the components related to the management of questions, the score, users...
GUI WEB	Package in charge of storing all the graphical user interface stuff
GameLogic	Component in charge of selecting questions, managing the movement and the score of every user.
QuestionManagement	Component that manages questions and answers
Views	Component including all the views of the application.
QuestionSelector	Component in charge of selecting the right questions
QuestionDownload	Component in charge of extracting the questions.
QuestionArray	This component stores the questions/answers
API REST	API Rest package
MovementManager	Movement component manager.
AcountManagement	Component that works with the accounts
API controller	API Rest controller
UserLogin	Manages the user verification to log in
API REST Interface	API Rest interface
ScoreManager	Component in charge of managing the scores

UserAdministrator	Manages the log in of the administrator

Details

DB

Name	Value
Description	Database package
Abstract	false
Leaf	false
Root	false
Visibility	public
Project Management	Name: Author — Value: Trivial_i1a Name: Create Date Time — Value: 07-may-2015 9:38:52 Name: Last Modified — Value: 10-may-2015 21:24:00

Children

Name	Description
DataBase	Database component

DataBase

Name	Value
Description	Database component
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false

Root	false
Indirectly Instantiated	true
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 07-may-2015 13:36:28 Last Modified / 10-may-2015 21:24:00

Relationships

Unnamed Usage	
To	QuestionManagerInterface
Visibility	Unspecified
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 09-may-2015 10:15:44 Last Modified / 09-may-2015 10:17:01

QuestionManagerInterface

Name	**Value**
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Leaf	false
Root	false
Stereotypes	Interface
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 09-may-2015 10:14:44 Last Modified / 09-may-2015 10:37:28

AccauntManagerInterface

Name	Value
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Leaf	false
Root	false
Stereotypes	Interface
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 09-may-2015 10:16:43 Last Modified / 09-may-2015 10:27:02

Relationships

Unnamed Realization	
To	N/A
Visibility	Unspecified
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 09-may-2015 10:18:04 Last Modified / 09-may-2015 10:27:02

Unnamed Usage	
From	N/A
Visibility	Unspecified
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 09-may-2015 10:18:09 Last Modified / 09-may-2015 10:27:02

PLAY FRAMEWORK

Name	Value
Description	Play environment
Abstract	false
Leaf	false
Root	false
Visibility	public
Project Management	Name: Author — Value: Trivial_i1a Name: Create Date Time — Value: 07-may-2015 9:43:38 Name: Last Modified — Value: 10-may-2015 21:24:00

Children

Name	Description
GUI WEB	Package in charge of storing all the graphical user interface stuff

LOGIC

Name	Value
Description	Logic package including all the components related to the management of questions, the score, users...
Abstract	false
Leaf	false
Root	false
Visibility	public
Project Management	Name: Author — Value: Trivial_i1a Name: Create Date Time — Value: 07-may-2015 9:38:17 Name: Last Modified — Value: 10-may-2015 21:03:58

Children

Name	Description
GameLogic	Component in charge of selecting questions, managing the movement and the score of every user.
QuestionManagement	Component that manages questions and answers
AcountManagement	Component that works with the accounts
AccountManagement	

GUI WEB

Name	Value
Description	Package in charge of storing all the graphical user interface stuff
Abstract	false
Leaf	false
Root	false
Visibility	public
Project Management	Name: Author — Value: Trivial_i1a Name: Create Date Time — Value: 07-may-2015 9:39:25 Name: Last Modified — Value: 10-may-2015 21:24:00

Children

Name	Description
Views	Component including all the views of the application.

Relationships

Unnamed Realization	
From	QuestionManagerInterface
Visibility	Unspecified
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:16:10 Last Modified: 09-may-2015 10:17:01

GameLogic

Name	Value
Description	Component in charge of selecting questions, managing the movement and the score of every user.
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:06:27 Last Modified: 10-may-2015 21:03:58

Children

Name	Description
QuestionSelector	Component in charge of selecting the right questions
N/A	
MovementManager	Movement component manager.
ScoreManager	Component in charge of managing the scores

N/A	

Ports

Name	Description
N/A	
N/A	

Resident Components

Name	Description
ScoreManager	Component in charge of managing the scores
QuestionSelector	Component in charge of selecting the right questions
MovementManager	Movement component manager.

QuestionManagement

Name	Value
Description	Component that manages questions and answers
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:02:25 Last Modified: 10-may-2015 21:13:59

Resident Components

Name	Description
QuestionDownload	Component in charge of extracting the questions.
QuestionArray	This component stores the questions/answers

Views

Name	Value
Description	Component including all the views of the application.
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	<table><tr><th>Name</th><th>Value</th></tr><tr><td>Author</td><td>Trivial_i1a</td></tr><tr><td>Create Date Time</td><td>07-may-2015 13:36:01</td></tr><tr><td>Last Modified</td><td>09-may-2015 10:37:28</td></tr></table>

QuestionSelector

Name	Value
Description	Component in charge of selecting the right questions
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	

Name	Value
Author	Trivial_i1a
Create Date Time	07-may-2015 13:44:23
Last Modified	10-may-2015 21:13:59

Relationships

Unnamed Association

To

Name	Value
End Model Element	MovementManager
Provide Property Getter Method	false
Provide Property Setter Method	false
Multiplicity	Unspecified
Visibility	Unspecified
Aggregation Kind	None
Navigable	Navigable
Derived	false
Derived Union	false
Read Only	false

	Static	false
	Leaf	false
	Type	MovementManager
	Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 09-may-2015 10:48:23
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 09-may-2015 10:48:23 Last Modified / 09-may-2015 10:51:55	

QuestionDownload

Name	Value
Description	Component in charge of extracting the questions.
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 07-may-2015 13:42:06 Last Modified / 10-may-2015 21:13:59

Relationships

<table>
<tr><th colspan="3">Generates : Association</th></tr>
<tr><td>To</td><td colspan="2"><table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>End Model Element</td><td> QuestionArray</td></tr>
<tr><td>Provide Property Getter Method</td><td>false</td></tr>
<tr><td>Provide Property Setter Method</td><td>false</td></tr>
<tr><td>Multiplicity</td><td>Unspecified</td></tr>
<tr><td>Visibility</td><td>Unspecified</td></tr>
<tr><td>Aggregation Kind</td><td>None</td></tr>
<tr><td>Navigable</td><td>Navigable</td></tr>
<tr><td>Derived</td><td>false</td></tr>
<tr><td>Derived Union</td><td>false</td></tr>
<tr><td>Read Only</td><td>false</td></tr>
<tr><td>Static</td><td>false</td></tr>
<tr><td>Leaf</td><td>false</td></tr>
<tr><td>Type</td><td> QuestionArray</td></tr>
<tr><td>Project Management</td><td><table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>Author</td><td>Trivial_i1a</td></tr>
<tr><td>Create Date Time</td><td>09-may-2015 10:42:08</td></tr>
</table></td></tr>
</table></td></tr>
<tr><td>Abstract</td><td colspan="2">false</td></tr>
<tr><td>Leaf</td><td colspan="2">false</td></tr>
<tr><td>Visibility</td><td colspan="2">Unspecified</td></tr>
<tr><td>Derived</td><td colspan="2">false</td></tr>
<tr><td>Project Management</td><td colspan="2"><table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>Author</td><td>Trivial_i1a</td></tr>
<tr><td>Create Date Time</td><td>09-may-2015 10:42:08</td></tr>
<tr><td>Last Modified</td><td>09-may-2015 10:47:29</td></tr>
</table></td></tr>
</table>

Relationships

Unnamed Usage	
To	API REST Interface
Visibility	Unspecified
Project Management	Name: Author — Value: Trivial_i1a Name: Create Date Time — Value: 09-may-2015 10:10:34 Name: Last Modified — Value: 09-may-2015 10:17:01

QuestionArray

Name	Value
Description	This component stores the questions/answers
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	Name: Author — Value: Trivial_i1a Name: Create Date Time — Value: 07-may-2015 13:42:50 Name: Last Modified — Value: 10-may-2015 21:15:09

Relationships

Generates : Association		
From	Name	Value

	End Model Element	QuestionDownload
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	Unspecified
	Aggregation Kind	None
	Navigable	Navigable
	Derived	false
	Derived Union	false
	Read Only	false
	Static	false
	Leaf	false
	Type	QuestionDownload
	Project Management	**Name** / **Value**: Author: Trivial_i1a; Create Date Time: 09-may-2015 10:42:08
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	
Project Management	**Name** / **Value**: Author: Trivial_i1a; Create Date Time: 09-may-2015 10:42:08; Last Modified: 09-may-2015 10:47:29	

API REST

Name	Value
Description	API Rest package
Abstract	false
Leaf	false
Root	false
Visibility	public

Project Management	Name	Value
	Author	Trivial_i1a
	Create Date Time	07-may-2015 13:35:14
	Last Modified	10-may-2015 21:24:00

Children

Name	Description
API controller	API Rest controller

MovementManager

Name	Value
Description	Movement component manager.
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true

Project Management	Name	Value
	Author	Trivial_i1a
	Create Date Time	07-may-2015 13:45:18
	Last Modified	10-may-2015 21:13:59

Relationships

Unnamed Association		
From	Name	Value
	End Model Element	QuestionSelector
	Provide Property Getter Method	false
	Provide Property Setter	false

	Method	
	Multiplicity	Unspecified
	Visibility	Unspecified
	Aggregation Kind	None
	Navigable	Navigable
	Derived	false
	Derived Union	false
	Read Only	false
	Static	false
	Leaf	false
	Type	QuestionSelector
	Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:48:23
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:48:23 Last Modified: 09-may-2015 10:51:55	

Unnamed Association		
From	**Name**	**Value**
	End Model Element	ScoreManager
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	Unspecified
	Aggregation Kind	None
	Navigable	Navigable
	Derived	false

	Derived Union	false
	Read Only	false
	Static	false
	Leaf	false
	Type	ScoreManager
	Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:48:32
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:48:32 Last Modified: 09-may-2015 10:51:55	

N/A

Name	Value
Service	true
Behavior	false
Conjugated	false
Derived	false
Is ID	false
Derived Union	false
Aggregation	Unspecified
Multiplicity	Unspecified
Read Only	false
Static	false
Leaf	false
Visibility	public
Project Management	**Name** / **Value** Author: Trivial_i1a

	Create Date Time	09-may-2015 10:16:34
	Last Modified	09-may-2015 10:27:02

Relationships

Unnamed Realization	
From	AccauntManagerInterface
Visibility	Unspecified
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:18:04 Last Modified: 09-may-2015 10:27:02

AcountManagement

Name	Value
Description	Component that works with the accounts
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:04:25 Last Modified: 10-may-2015 21:24:00

Children

Name	Description
UserLogin	Manages the user verification to log in
UserAdministrator	Manages the log in of the administrator

Resident Components

Name	Description
UserLogin	Manages the user verification to log in
UserAdministrator	Manages the log in of the administrator

API controller

Name	Value
Description	API Rest controller
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	Name: Author — Value: Trivial_i1a Name: Create Date Time — Value: 07-may-2015 13:35:31 Name: Last Modified — Value: 10-may-2015 21:24:00

UserLogin

Name	Value
Description	Manages the user verification to log in
Active	false

Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 07-may-2015 13:45:50 Last Modified: 10-may-2015 21:24:00

Relationships

Unnamed Association

To	**Name** / **Value** End Model Element: UserAdministrator Provide Property Getter Method: false Provide Property Setter Method: false Multiplicity: Unspecified Visibility: Unspecified Aggregation Kind: None Navigable: Navigable Derived: false Derived Union: false Read Only: false Static: false Leaf: false Type: UserAdministrator Project Management: **Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:47:43
Abstract	false

Leaf	false
Visibility	Unspecified
Derived	false
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 09-may-2015 10:47:43 Last Modified / 09-may-2015 10:51:55

API REST Interface

Name	Value
Description	API Rest interface
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Leaf	false
Root	false
Stereotypes	Interface
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 07-may-2015 13:40:36 Last Modified / 09-may-2015 10:17:01

ScoreManager

Name	Value
Description	Component in charge of managing the scores
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false

Root	false
Indirectly Instantiated	true
Project Management	**Name** / **Value** Author: Trivial_i1a Create Date Time: 07-may-2015 13:45:30 Last Modified: 10-may-2015 21:13:59

Relationships

Unnamed Association

To	**Name** / **Value** End Model Element: MovementManager Provide Property Getter Method: false Provide Property Setter Method: false Multiplicity: Unspecified Visibility: Unspecified Aggregation Kind: None Navigable: Navigable Derived: false Derived Union: false Read Only: false Static: false Leaf: false Type: MovementManager Project Management: **Name** / **Value** Author: Trivial_i1a Create Date Time: 09-may-2015 10:48:32
Abstract	false
Leaf	false
Visibility	Unspecified
Derived	false
Project Management	**Name** / **Value**

	Author	Trivial_i1a
	Create Date Time	09-may-2015 10:48:32
	Last Modified	09-may-2015 10:51:55

Relationships

Unnamed Association		
From	**Name**	**Value**
	End Model Element	ScoreManager
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	Unspecified
	Aggregation Kind	None
	Navigable	Navigable
	Derived	false
	Derived Union	false
	Read Only	false
	Static	false
	Leaf	false
	Type	ScoreManager
	Project Management	**Name** / **Value**: Author: Trivial_i1a; Create Date Time: 09-may-2015 10:51:01
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	
Project Management	**Name**	**Value**

	Author	Trivial_i1a
	Create Date Time	09-may-2015 10:51:01
	Last Modified	09-may-2015 10:51:55

UserAdministrator

Name	Value
Description	Manages the log in of the administrator
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true
Project Management	**Name** / **Value** Author / Trivial_i1a Create Date Time / 07-may-2015 13:46:16 Last Modified / 10-may-2015 21:24:00

Relationships

Unnamed Association		
From	**Name**	**Value**
	End Model Element	UserLogin
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	Unspecified
	Aggregation Kind	None

<table>
<tr><td></td><td><table>
<tr><td>Navigable</td><td>Navigable</td></tr>
<tr><td>Derived</td><td>false</td></tr>
<tr><td>Derived Union</td><td>false</td></tr>
<tr><td>Read Only</td><td>false</td></tr>
<tr><td>Static</td><td>false</td></tr>
<tr><td>Leaf</td><td>false</td></tr>
<tr><td>Type</td><td> UserLogin</td></tr>
<tr><td>Project Management</td><td><table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>Author</td><td>Trivial_i1a</td></tr>
<tr><td>Create Date Time</td><td>09-may-2015 10:47:43</td></tr>
</table></td></tr>
</table></td></tr>
<tr><td>Abstract</td><td>false</td></tr>
<tr><td>Leaf</td><td>false</td></tr>
<tr><td>Visibility</td><td>Unspecified</td></tr>
<tr><td>Derived</td><td>false</td></tr>
<tr><td>Project Management</td><td><table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>Author</td><td>Trivial_i1a</td></tr>
<tr><td>Create Date Time</td><td>09-may-2015 10:47:43</td></tr>
<tr><td>Last Modified</td><td>09-may-2015 10:51:55</td></tr>
</table></td></tr>
</table>

User´s system manual

General functionality

This version of the application have a more visual interaction to the user and the program will be divided into several phases:

- At first a logging window will appear in which the user will use his username and password to differentiate between administrator or common user, this username and password will be compare in a txt document.

- Then the game begins, a squared board will appear in which the game will be played, a dice will appear and the available boxes will appear.

- Player chooses a square and a dialog with the question of that category is shown, in which you have to choose one answer, and the game will be elapsed in this way till you reach the center of the board.

- Finally when you reach the center one question of each theme, if you answer correctly the game finishes and you win.

The load of the questions file is done in a automatically way.

GIFT format

// text	Comment until end of line (optional)
::title::	Question title (optional)
text	Question text (becomes title if no title specified)
!theme!	Question Theme(optional)
{ ... =right ... }	Correct answer for multiple choice, (multiple answer? -- see page comments) or fill-in-the-blank
{ ... ~wrong ... }	Incorrect answer for multiple choice or multiple answer