**University of Oviedo**

**Documentation Second Derivable**

**Software Architecture**

**Ana Areces**

**Raquel Arrojo**

**David Casado**

**María González**

**Álvaro Palanco**

**Iván Sánchez**

Contents

[Project information 2](#_Toc416475567)

[Architecture diagrams 3](#_Toc416475568)

[Package view 3](#_Toc416475569)

[Component views 4](#_Toc416475570)

[Extract 4](#_Toc416475571)

[Game 5](#_Toc416475572)

[Deployment view 6](#_Toc416475573)

[User manual 7](#_Toc416475574)

[Initial window 7](#_Toc416475575)

[About 8](#_Toc416475576)

[Create a new user 8](#_Toc416475577)

[Play 9](#_Toc416475578)

[List Users 9](#_Toc416475579)

[Statistics 10](#_Toc416475580)

[System Manual 11](#_Toc416475581)

[System requirements 11](#_Toc416475582)

[Project information 11](#_Toc416475583)

[Logic 11](#_Toc416475584)

[GUI 12](#_Toc416475585)

# Project information

All the documentation files can be found inside the documentation folder in our Github repository.

<https://github.com/Arquisoft/Trivial_i1b>

This is a direct access to the Readme that contains the general information of the group.

<https://github.com/Arquisoft/Trivial_i1b/blob/master/README.md>

This folder contains the Documentation information like manuals and all the diagrams, also the Visual Paradigm project.

<https://github.com/Arquisoft/Trivial_i1b/tree/master/Documentation>

This one is the Eclipse project, where you could find the src folder with all the implementation.

<https://github.com/Arquisoft/Trivial_i1b/tree/master/Game>

The cucumber tests can be found following this link.

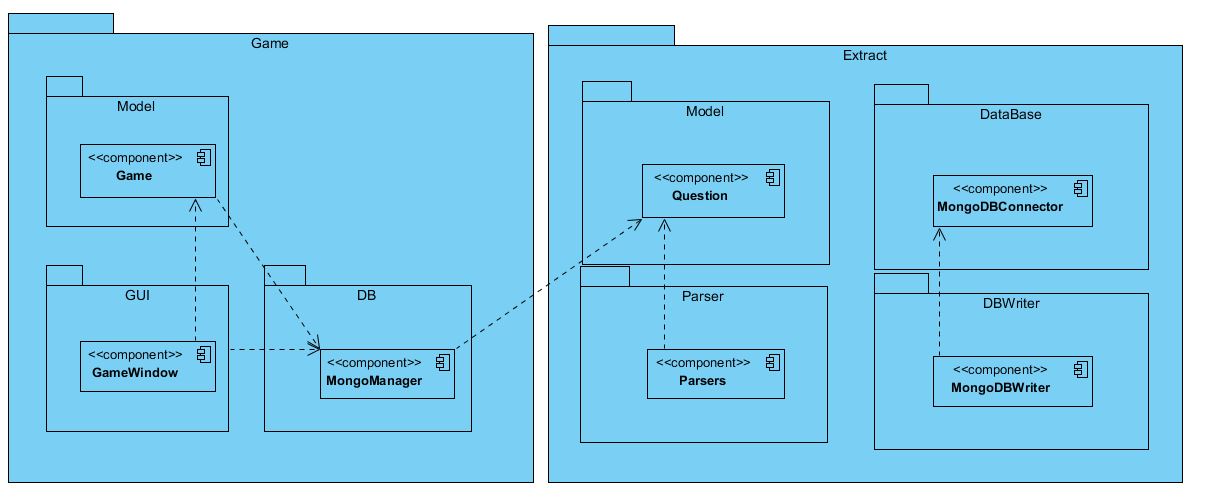
<https://github.com/Arquisoft/Trivial_i1b/tree/master/Game/src/test/java>

The generated documentation.

<https://github.com/Arquisoft/Trivial_i1b/blob/master/Documentation/GeneratedVPDocumentation.pdf>

# Architecture diagrams

## Package view

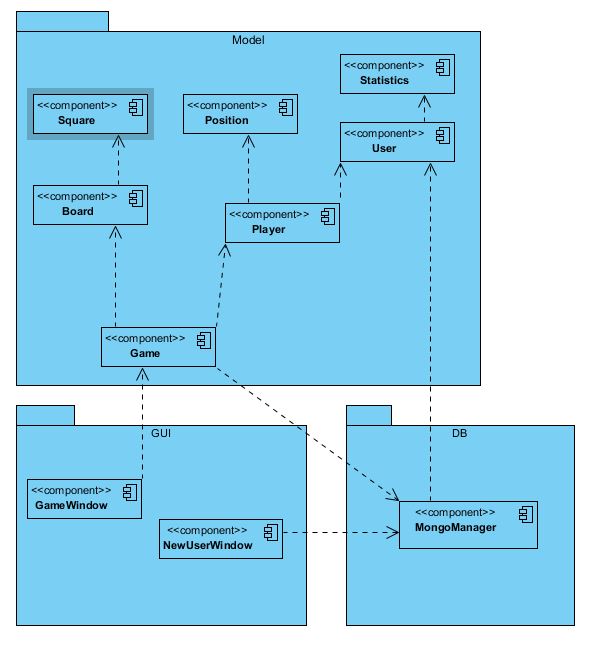


## Component views

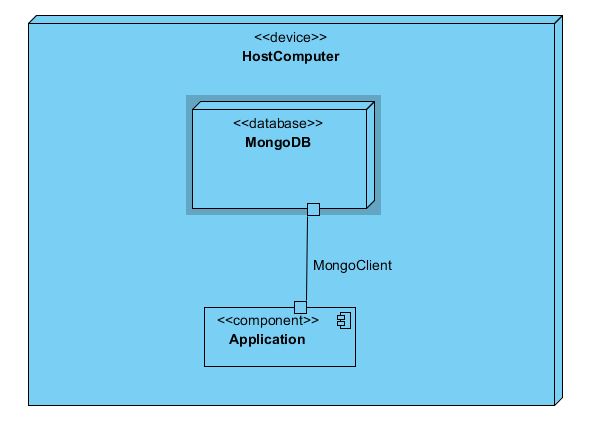
### Extract



### Game



### Deployment view



# User manual

This is a web application that works locally but can be uploaded to a server; since it is based on the original Trivial game everybody will be able to learn how to use it in almost no time.

The most important thing the user must know before starting to use or application is that it is composed by 5 different pages:

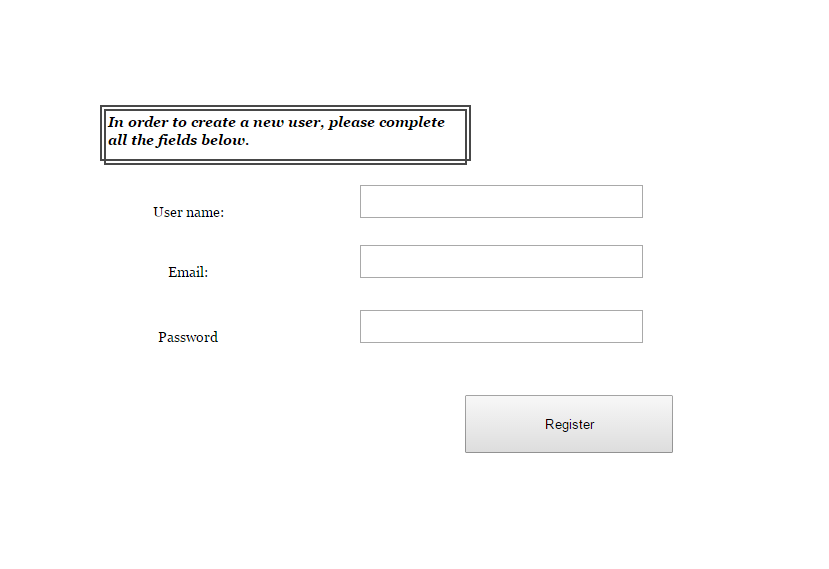
## Initial page

The first window that appears when the application is ran. In this window, you can change the background color with the buttons on the left, go to the page where you can register as a user or go to the login page.

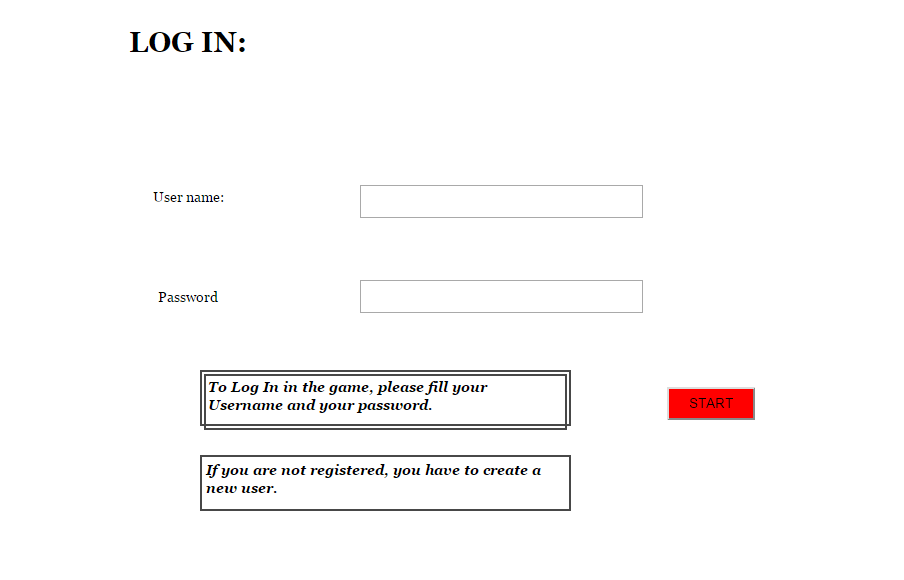


## Create a new user

If you want to create a new user, you must provide some personal data like your username, your email and a password. All this information is required to create the user and will be saved to the database after you click in the register button.

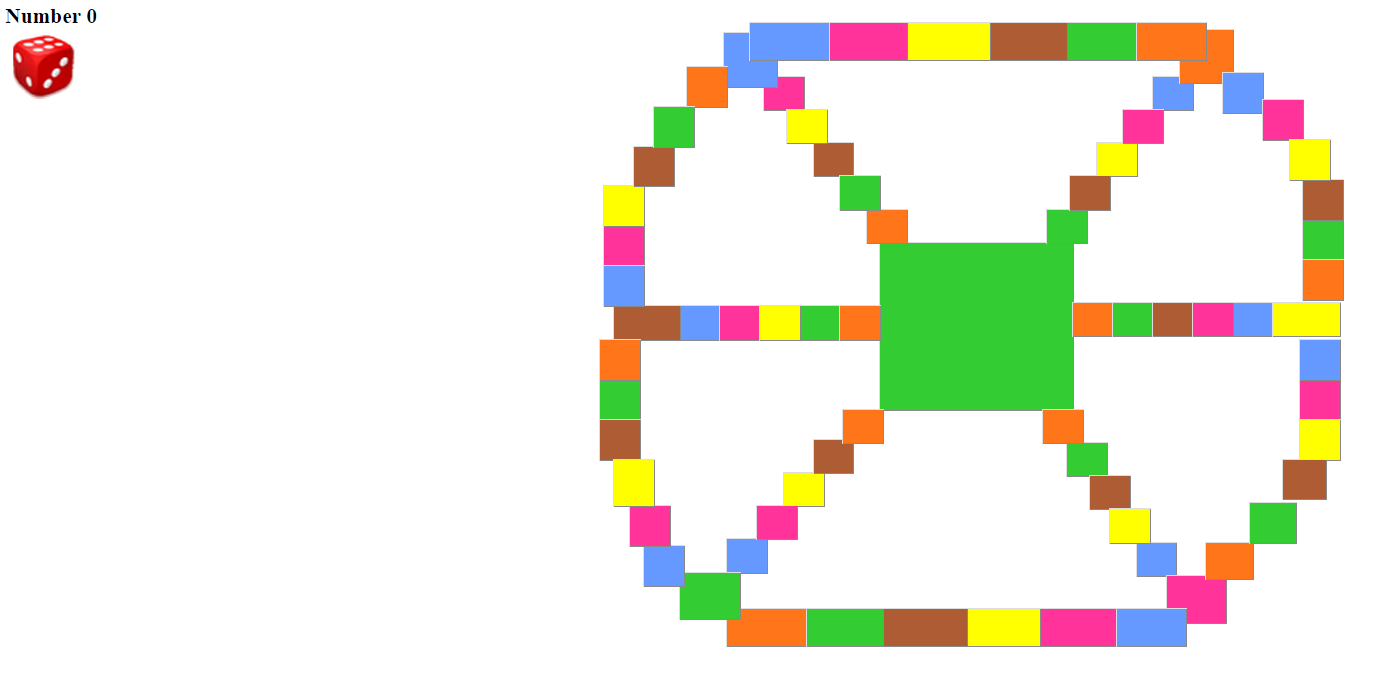


## Login

In this window you can login with your username and password you previously created in the register page. This information is going to be checked when you click in the start button, after that if the information is correct the play page is going to be shown. 

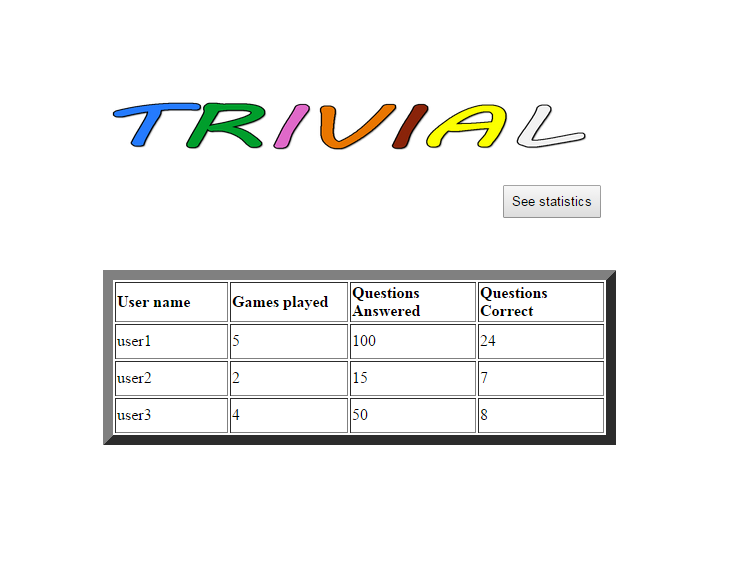
## Play

This is the main window of the application. Here is the board that should show where the player is. To start the game, the dice has to be pressed and the player has to decide which category he/she prefers. Depending on the category selected, one question or another will be retrieved from the database and asked to the player.



## Statistics

In this page the user will be able to see a small piece of information as a table. The information shown is the username, number of games played, number of answers and number of correct ones for all players.



# System Manual

## System requirements

In order to work properly with this project the main requirement is use a Java version greater than or equal to Java 1.7, otherwise some compliance errors may appear in the project.

## Project information

The project is composed by two different packages, one for the logic of the game and another one for the user interface.

In the following sections we’ll explain the main classes contained in those packages.

### Logic

***MongoQuestionManager***

The one which controls the questions in the database. It has two constants, one containing the name of the database and another one with the collection. In the constructor the connection to the database is created. This class has to methods. One of them is private and it is used in order to translate a Document object to a Question one. The other one is public and returns all the questions in the database.

***MongoStatisticsManager***

It has a public method that, given a user saves its statistics in the database. There are other two methods, one of them in order to retrieve the statistics of a certain user passed as parameter; and the other is used when it is necessary to update the statistics. In this last case, the user is needed in order to update the correct one.

***MongoUserManager***

This class it is important because it manages all related to users in the database. There is a method in order to retrieve a certain user by its name but we also have implemented another method in order to get all the users in the database.

It has also a method in order to update the information of a certain user.

***Board***

This class represents the board and initializes it. It allows the movement through it depending on the result obtaining when dropping the die.

***Die***

It simulates dropping the die. Its unique method return a number from 1 to 6.

***Player***

The function of this class is create players to associate them to the database each time a user is logged in.

***Position***

This class provides the information about in which cell the player is. The method walk returns the array in which the player is placed, in case the returned value is 0 its position is in one of the cell in the external circle, otherwise is in one of the paths arriving to the center.

***Statistics***

This class retrieves the statistics for each user in the database. All the methods in this class are getters and setters.

***User***

Simulates the users in the database, the identification of each user is the login that is unique.

***Game***

This class is the factory to be called from the interface. From it you can access all the methods needed in order to play, such as creating the players, move them …

***SimpleSquare***

It represents the cell in the external part of the board. In order to create them a position and a category is needed.

***CompositeSquare***

In this case, the cells to be represented are the internal paths. It contains a list with more squares representing the cells contained in each path.

### GUI

We will not explain the complete implementation of the class in this package. Since they are classes representing the user interface it seems to be enough to explain the main purpose of each one of them.

***About Trivial***

This class is used in order to give the user the information about the development team.

***Game Window***

As it name says, is the one in charge of providing the space for playing the game.

***Initial Window***

The one the user see the firs when starting the application. From this one you can access to the others.

***List players***

This class will be used in order to show the administrator the users that have played the game. From this window he will be able to select one of them and see the statistics for the selected one.

***New User Window***

In order to play the game is it compulsory to have the required player saved in the database. In case it is not, the application provides this window in order to allow the user to add a new player to the database.

***Statistics Window***

The last window of the user interface shows the statistics of the player selected in the List Players window.