



Nerderly Assessment

DECEMBER 23, 2013 / NAT JAVA VERSION 1.3.1

PHONE (877) 664.6373
FAX (952) 948.1611
EMAIL info@nerderly.com
WEB <http://nerderly.com>



NAT SPECIFICATION

XBOX GAME VOTING APPLICATION

DOCUMENT OBJECTIVE

The purpose of this document is to provide detailed documentation that clearly defines the application that an applicant will create for the Nerderly Assessment Test.

NOTES TO CHALLENGER

The Nerderly Assessment Test (NAT) is a chance for you to show off your development capabilities and prove your experience. While this is a relatively simple web application, we expect you, the applicant, to think of the end result as a finished project that we would eventually launch for internal use. It should be maintainable, and reasonably free from errors and defects. We don't expect a high-design, high-security/high-performance driven application.

You should spend upwards of 10 hours on this challenge. We understand that this is typically not enough time to develop a fully robust application. You may feel free to spend additional time on it if you wish, but just know that it's not necessary to do so as long as you meet the spec's requirements. The primary purpose of the NAT is to demonstrate your mastery of Java development; So ,while JavaScript may be used to enhance the user experience of the site, all functional requirements should be implemented in Java. Additionally, the submission should be written in the Java language, not alternative JVM languages, such as Groovy or Scala. Due to the time goal and simplicity constraints, if you make a development decision that goes against what you would normally perform, we encourage you to include notes with your submission.

If you have any questions regarding the spec, please feel free to email your candidate advocate or the developer assigned to you for questions. We will be able to answer general questions about the challenge or make clarifications about the spec, but cannot offer any specific advice for completing it.



APPLICATION DESCRIPTION

The Nerderdy needs an application to track the interest in new games for our Xbox 360. This application will display the games we currently own, the games we would like to buy, and the number of votes for each game. Each employee at The Nerderdy should be able to vote for their favorite game **or** add a new game to this list once per day. At the end of the week, if we reach our productivity goals the game with the most votes will be purchased and marked as a game we currently own.

A database will be created to store the game and vote information. A user interface will be developed to display the games we own, display games we are voting on, and allow users to submit new titles for voting, and designate a game as owned.

END USER DESCRIPTION

The end users of this application will be the programming staff at The Nerderdy. The majority of these users prefer Google Chrome release channel for a web browser and are running Windows 8 Professional for an Operating System. It can be assumed that the programmers will only vote for their favorite game from their workstation on the LAN at The Nerderdy headquarters, so setting a cookie to determine whether or not the user has voted is an acceptable low tech solution.

WEB APPLICATION REQUIREMENTS

The details for this web application are defined below.

DISPLAY GAMES

The user interface portion of the application will retrieve the games data via the database defined in the Database Specification section below. Games will be separated into those that we currently want, and those than we own. The games that we want will display the current vote count, and must be sorted by this vote count in descending order. The games that we own must be sorted alphabetically, and will not display the vote count.

VOTE FOR A GAME

The user interface will provide the user with an ability to vote for any Xbox game that The Nerderdy does not currently own, based on the restrictions defined below in "Voting and Add New Title Restrictions".

ADD NEW TITLE/GAME

If an Xbox game is not currently in the list of games we want or own the user will have the ability to add that game to the list by providing the game's title. There will be no validation that the title is an actual Xbox game but to keep the list clean duplicate titles will not be allowed.

Adding a new game will also automatically add the first vote on behalf of the user.

VOTING AND ADD NEW TITLE RESTRICTIONS

Users must only be allowed to submit one vote **or** add one new game title per day (midnight to midnight). Additionally, voting or adding a new game title must be prohibited on Saturday and Sunday.

MARKING A GAME AS OWNED

A separate page will exist to allow a user to indicate a game that we want is now owned. There will be no restrictions as to how many games can be owned.

CODE EXPECTATIONS

The developer is expected to adhere to the following guidelines:

WELL DOCUMENTED

The application code will be created with the assumption that other developers will work on it in the future, so it should be readable and commented enough to make it easy for others to understand and maintain.



FRAMEWORK AND TOOLSET

Where possible, the application code should utilize APIs built-in to the Java platform, as well as your selected web framework to perform tasks. The application code may pull in any third party libraries or tools, open or closed source, as needed, as long as they do not require a paid license.

CODING STYLE

Code style should be consistent throughout the application and conform to industry accepted style conventions for the given language. The application code should be developed using the latest version of the language and should adhere to modern programming practices and style conventions.

OBJECT ORIENTED

The code produced is expected to follow industry accepted object oriented principles and patterns.

ERROR HANDLING

All application errors must be handled gracefully and display user-friendly error messages if necessary. Submission should avoid common web security vulnerabilities and exploits.

HTML CODING STYLE

The quality of the site design will not be judged, but the HTML must be semantic and valid.

DATABASE SPECIFICATIONS

The application must use a relational database, such as MySQL, to store the games and votes. The following is the recommended minimum database pseudo-schema.

GAMES TABLE

id - integer

title - text (Title of Game)

owned - boolean (default = false)

created - timestamp

VOTES TABLE

gameId - integer

created - timestamp