Rps Use Case

Document Information

|  |  |
| --- | --- |
| **Document Title** | Rps |
| **Document Owner** | Jeremy Blake |
| **Version** | 1.0 |
| **Status** | Final |
| **Date** | April 8, 2020 |

1. Brief Description

The basic use for the software is to be able to play the game of roshambo or rock,paper,scissors. The program starts when the interface is presented to the end user, and stops when the customer exits the program.

1. Actors

The following roles and systems are involved with the program Rps use case:

* User
* Interface presenting the buttons and results
* A backend Tomcat server integrated with a MariaDB database

1. Pre-Conditions

The following conditions must be validated by the system to be true before the Rps functionality begins:

* The system must power on
* The Rps program must be running

1. Basic Flow

The basic flow or happy path for the Rps program:

1. The system is on and the interface is presented to the user
2. The user selects any of the top 3 buttons of rock, paper, or scissors
3. The computer response as well as the game results are then generated and displayed in the interface below the buttons, most recent result will always be on top
4. The results will contain what the user played, what the computer played, the round number, the who won message, as well as a current tally of the wins for the user or the computer for the current session
5. After the requested number of play-throughs have been reached, the results of the play-throughs can be saved to a JSON export file, or they can be viewed through a Tomcat portal, that will list the most recent play-throughs
6. The user selects the ‘x’ in the top right corner and the program execution terminates
7. Happy path user case ends
8. Alternate/Exception Flows

1a. The system is not powered on, or the interface is not presented

1b. The system must be powered on, or the program must be started

1c. Normal use case begins

2a. There are 4 total buttons in the Rps interface that can be selected by the user, the selection of any of them continues the normal use case of the happy path

2b. If ‘x’ is selected at this stage, the program terminates and the use case ends

3a. There is no additional input available to the user.

4a. There is no additional input available to the user.

5a. The recent play-throughs can be overwritten, the JSON file must be archived away if the current play-through is to be preserved

1. Post Conditions

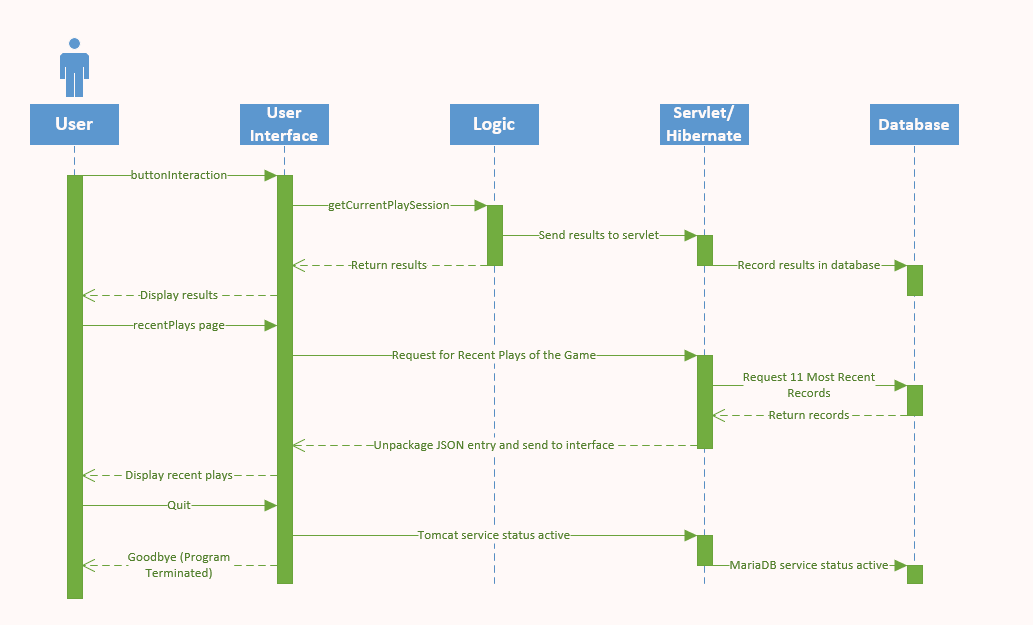
The following condition must be validated by the system to be true before the Rps functionality ends

* The ‘x’ button can be selected at any time for the program to terminate
* A break command is executed in the middle of the program

1. Supplemental Requirements

Depending on future requirements, additional buttons and functionality could be added. The interface is generally rudimentary, but the simplicity is what makes the minimization of the sad path possible. Additional tweaking of the game’s logic could be possible, but the ability to overcome random spamming of user choices is already present in the algorithm.

1. Visual Model



Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V. | Date | Author | Description | Status |
| 1.0 | 8-APR-2020 | Jeremy Blake | Use case details entered in full | Final |
|  |  |  |  |  |