**Document 02 – Sprint Report**

Contents

[1 Document Ownership 1](#_Toc135656804)

[2 Class Diagram 2](#_Toc135656805)

[3 Retrospective 2](#_Toc135656806)

[4 Mid Sprint Demo 2](#_Toc135656807)

[5 Video Demo 3](#_Toc135656808)

# Document Ownership

This document is contained in your GitHub repository in a folder named *docs*. At the end of the Sprint, Sections 2 and 3 will contain information you supply.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Team |  | | Team Member Names |  |
|  | |  |  | | --- | --- | | 1. |  | | 2. |  | | |  |  | | --- | --- | | 3. |  | | 4. |  | | 5. |  | |

Video Link(s) (See [Section 5](#_Video_Demo)):

|  |
| --- |
|  |

# Class Diagram

|  |
| --- |
| **Deliverable**  At the conclusion of this sprint, you should make a class diagram using StarUML. You can make multiple diagrams at different levels of granularity, or just break it up. **The diagram should be included in this document along with several paragraphs to explain your design.** The diagram(s) must be readable. Also, include the diagrams saved as image files (jpg or png) in your *docs* folder on GitHub. |

# Retrospective

|  |
| --- |
| **Deliverable**  Towards the end of the sprint, read this short page about what a software retrospective is and why it is important and then answer the questions below. **The answers should be included in this document**.  <https://searchsoftwarequality.techtarget.com/definition/Agile-retrospective> |

Meet as a group and discuss the following questions and provide a group written response below. Write as much as is appropriate.

1. What worked well for us?

|  |
| --- |
| **Answer** |

1. What did not work well for us?

|  |
| --- |
| **Answer** |

1. What actions can we take to improve our process going forward?

|  |
| --- |
| **Answer** |

# Mid Sprint Demo

|  |
| --- |
| **Deliverable**  By Friday, July 7, have someone in your group make an appointment with me following the directions on the Schedule. As many team members as are available should attend this meeting. Requirements:   * Cameras are turned on * Your IDE’s font size should be set to 16 (in Eclipse: Window, Preferences, General, Appearance, Colors & Fonts, Basic, Text Font (at bottom of Basic), Edit) * Increase the zoom on your browser to 125%   To begin, I’ll want to see various artifacts listed below. Show them, and I’ll ask any questions I have. You don’t need to prepare any remarks:   * Project Board in GitHub * User Stories spreadsheet: User Stories Tab (columns H, I, J), System Tests Tab – show a completed one. * Class diagram of code base * Open your code in an IDE. I’ll ask questions.   Next:   * Briefly demo your code. * What questions do you have for me? |

# Video Demo

|  |
| --- |
| **Deliverable**  When your project is complete, create a video that demo’s your User Stories and provide the link in [Section 1](#_Document_Ownership). Requirements:   * You can make a single video, or, if needed, 2 or 3. * The total length should be whatever is needed to accomplish the agenda below. * Preferably, post your video(s) on YouTube. * Agenda:  1. (3-5 minutes) Explain your design. You should use class diagrams sized so that the relevant portions fill as much of the screen as possible. As part of this, explain how your design implements MVC. 2. (1-2 minutes) Choose one User Story and step through the code as if it were being executed. You’ll start by showing the code where the appropriate event handler responds to the user. Next, to whatever it calls, etc, explaining as you go. 3. (1 minute) GitHub: 4. Display your Project Board for 10 seconds. 5. Display the Issues for 5 seconds, then display the “Closed” issues 6. For each User Story: 7. State the number of the User Story, and then state the User Story itself. 8. Demonstrate it with your software. If you have multiple system tests, you can demo all of them, or just the main (success) one. You can decide on the basis of time that you have. |