**Document 04 – Project 2 Wrap-up**

**This document is contained in your GitHub repository in a folder named *docs*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Group | 1 | | Group Member Names |  |
|  | |  |  | | --- | --- | | 1. | Ryan Slaybaugh | | 2. | Luke Edmondson | | |  |  | | --- | --- | | 3. | René Zambrana | | 4. | Parker Harris | |
|  |  |  |

1. **Video Demo**

|  |
| --- |
| **Deliverable**  Create a video for your project presentation. The agenda for your demo:   1. (1-3 minutes) Describe your project 2. (3-5 minutes) Discuss the design of your system using a class diagram.  * Explain at a high-level what each (or the most important) classes’ responsibilities are. Your goal is to give me a feel for your architecture and how the pieces fit together. * Don’t read off the list of methods! You can mention some key methods, or just describe what responsibilities each class has. If you need to go into more detail, it is fine, to show portions of the design and explain. * The class diagram should **not** show instance variables for any of the classes (in ObjectAid, select the class, right-click, and choose: *Show Attributes, None*) * The class diagram should **not** show methods for any Gui classes (in ObjectAid, select the class, right-click, and choose: *Show Operations, None*)  1. (3-5 minutes) Choose two tests (or sets of related tests) that you think are good tests. These can be unit or integration tests. Then, for each:  * Is this a unit or integration (or other) test? * What method(s) is being tested? * In what class(es)? * What does the method(s) do? * How did you test it? How did you decide on the inputs? * Did you use jUnit? * Any other comments?  1. (5-10 minutes) Pick a set of user stories to demo. Then, for each:  * Display a user story (text, in word or whatever) and read it, Expand on it if necessary. * Illustrate how the software fulfills it.   You may break this into 3 videos if you like. Just put the links here. |

|  |  |
| --- | --- |
| Link | <https://valdostaedu-my.sharepoint.com/:f:/g/personal/rledmondson_valdosta_edu/EjeChkcbqgNEstE7HSLwhbsBgUANHTvmQ3uj4TUNsVqhfg?e=3zr15C> |

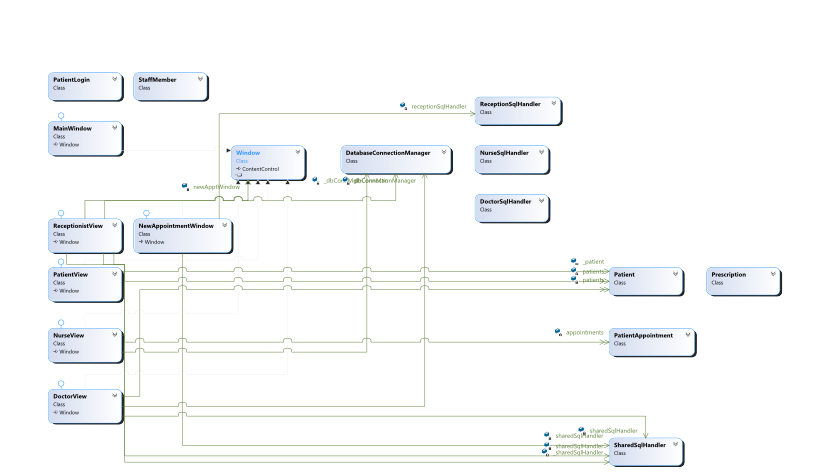
1. **Development Status**

|  |
| --- |
| **Deliverable**  Provide a table of user stories that are completed, and another table off stories that are in progress. |

|  |  |  |
| --- | --- | --- |
| **Num** | **Completed User Story Title** | **Comments** |
| 1. | Patient: View Appointments |  |
| 2. | Receptionist: Patient Balance | https://github.com/vsu-se/Team-1-Fall-20/issues/30 |
| 3. | Receptionist: Patient Balance | https://github.com/vsu-se/Team-1-Fall-20/issues/31 |
| 4. | Receptionist: Patient Signin |  |
| 5. | Doctor: Patient Medical History |  |
| 6. | Doctor: Prescribing Medicine |  |
| 7. | Receptionist: Add/Edit Appts |  |
| 8. | Receptionist: Calandar |  |
| 9. | Patient: View Prescriptions |  |

|  |  |  |
| --- | --- | --- |
| **Num** | **In-Progress User Story Title** | **Comments** |
| 1. | Patient: Payment |  |
| 2. | Nurse: Edit/Update Patient Info | This functionality was partially complete, but needed additional tuning. |
| … |  |  |

1. **Class Diagram**



|  |
| --- |
| **Deliverable**  Use Object Aid (or some other software) to develop a neat, legible, properly sized UML class diagram(s) showing your current design that exactly reflects your code. If your system is large, it is better to show a diagram with just the classes and their associations. Then, follow that with 2 or more diagrams showing the details of each class. You can use the class diagram from the demo. |

1. **Retrospective**
2. Meet as a group and discuss the following questions and provide a group written response below:
3. What worked well for us?

|  |
| --- |
| **Answer:** Working individually on one view each allowed us to avoid some conflicts initially and let each individual show their own creativity with the project. Using GitHub allowed for efficient integration of code between all members present, allowing conflicts to be resolved more easily and to share code as well. |

1. What did not work well for us?

|  |
| --- |
| **Answer:** Having to meet online reduced the number of times that we met, as we weren’t able to see each other on campus with conflicting schedules. This led to slower progress than would have been made if meetings were done more often. Working individually also led to an initial repetition of shared code that later had to be fixed by creating classes that could be used by all. |

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

|  |
| --- |
| **Answer:** We could spend more time working on diagrams for our project so that we are all on the same page. We can then come up with code issues that need to be resolved, rather than using an oversimplified user story. Basically, break the code down into chunks and assign users for them. |