Post Project Review

Smart Roster

**Retrospective framework**

Our team utilized the “Three L’s” retrospective framework for our retrospective. This framework gives easy, understandable, and straightforward details and information. We have gathered the most common input and keywords for each L.

**Three L’s**

Liked:

* Work impacts people in the future
* Everyone in the team in communicating efficiently
* The project pieced very well
* Our communication with our clients were smooth and valuable
* Making connections with other team members despite never having met in real life

Learned:

* Different technologies and methods
* How nurses and patients are rostered
* Nurse variables and terms
* Time management
* General scheduling process of a nurse

Lacked:

* Code quality (There are a lot of edge cases that our app can’t handle)
* Code tidiness (Code could be reformatted even more)
* We weren’t able to deploy to the original timeline
* Scope creep
* We didn’t let the Client have enough playtime with our app while developing

**Project Summary**

Our client approached us with a unique problem which was with their current nurse scheduling system, every shift, patients require care from nurses at an appropriate skill level and at present nursing assignments are completed manually by a Clinical Nurse Leader (CNL) or delegated Charge Nurse (CN) which takes up too much time and energy from the CNL or CN. In addition, the current scheduling system is based on using a pen and paper, which can likely result in human error when assigning patients.

Our goal was then to make scheduling more efficient, consistent, and reduce the amount of steps within the process as well as removing the need to manually perform this process. The main factor of their problem is time, and making a product that makes their time more efficient will bring great value in the long run.

We have met our project goals and provided a solution to the problem statement that we were given. The main goal was to eliminate the hassle of a fully manual, handwritten nurse roster. Since this project was given to us with a lot of flexibility in what technologies and how we design our solution, we have provided a web-based application with a primary function of automating nurse-patient assignments quickly and efficiently, solving our problem statement. We have also given the nurses a manual override feature to make changes to the roster that fits their special needs or situations.

**Project Roadmap**

|  |  |  |
| --- | --- | --- |
| Milestone 1 (Kickoff/Research) | Sprint 0 | * UX/UI: Wireframing and Design * Database: Basic Authentication |
| Milestone 2 (development push)  Backbones scheduler | Sprint 1 | * Algorithm: Extract data from input to use for algorithm, patient assignments * UX/UI: Prototype in progress |
| Sprint 2 | * Algorithm: Pod assignments based on hard constraints * UX/UI: High-fidelity Prototype complete * Database: Authentication for CNL/CN and RNs established |
| Sprint 3 | * Goal: Super barebones scheduler with automation * UX/UI: Basic Front-End mapping * Algorithm: Introduce soft constraints |
| Milestone 3 (development and testing)  Improved Automated Scheduler | Sprint 4 | * Algorithm: Continue implementing soft constraints * UX/UI: HTML/CSS |
| Sprint 5 | * Algorithm: Manual Override * UX/UI: CSS |
| Sprint 6 | * Algorithm: Improve algorithm and output data to platform * UX/UI: Javascript |
| Milestone 4 (testing and deployment) | Sprint 7 | * Algorithm: Last minute debugs and/or changes |
| Sprint 8 | Testing of Deployment and System |
| Sprint 9 | Polish project and Hand-over |

We as a group were able to stick to our road map pretty well. The biggest change we had to make was that as we progressed in our project, our client realized the scope for what they wanted deployed was greater than what they originally expected. Because of this, in the later half of the project we were able to not have to worry about making sure the deployment went well and could focus on the additional features that they wanted.

**Project Deliverables (Planned vs. Actual)**

Original Planned Deliverables:

|  |  |
| --- | --- |
| **Deliverable** | **Description** |
| Nurse Scheduling Algorithm | A working algorithm that automates nurse-patient assignments. We will also give the source code of the algorithm. |
| Web Application | User Friendly interface for users to interact with the system |
| Manual Override and Constraints | If an user manually overrides the current pod/patient-nurse assignment, there will be an alert/prompt |
| Documentation | We will log all development and process everything into documentation for the client to review and look back on. |
| Application Deployment | We will make sure everything works and is usable by the client |

Actual Outcome:

|  |  |
| --- | --- |
| **Deliverable** | **Description** |
| Nurse Scheduling Algorithm | We have a working algorithm, but only finds the local optimum and it can not handle edge cases. Our code has been reformatted and optimized slightly and could be further reformatted and optimized. |
| Web Application | Our interface has been designed to be similar to their pen-paper sheets that they manually fill out, giving them a familiar feel and a friendly interface. |
| Manual Override and Constraints | Manual override has been completed, but there are a few features that the Client would like to be implemented. We have rejected their suggestions due to scope creep. We have also completed alerts/flagging how we intended. Constraints have been implemented. |
| Documentation | We actively documented each sprint(week) and also documented our meetings. Code comments could have used some more work in other areas of the application and not just the algorithm. |
| Application Deployment | Due to how much the scope increased, the client decided that another development ISSP session would be needed and so deployment would be delayed till the scope can be covered |

**Transition to Operations**

Our biggest challenge was that the scope of the project became simply too much for our limited time and we were ultimately not able to deploy to operations. We were however able to show them a working MVP which they were pleased with that they were able to test and play with. Once they let us know that we would not be actually deploying our project, our focus switched to making sure the hand over of the project to the next group would go smoothly. We made sure our code is commented and makes sense, we have a proper README file which explains how to run the program as well as the dependencies it has, and we also have a hand-off document detailing the project process and methodology our team went through.

**Risks and Issues**

Due to the nature of our project. The main risk for us was handling sensitive data. As medical records and other relevant information is considered private, we have to be careful how we build the application and take privacy as a big factor. Another issue that we ran into was scope creep. Near the end of the project, our client began inquiring about features that could be added. Which were not initially mentioned or planned. This was likely the result of communication error, or simply seeing potential later in development that had caught their eye. We attempted adding the features we thought we could implement as ‘Stretch Goals’. Although it was not ideal.

# Project Risk Register

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk Description** | **Risk Impact** | **Risk Probability** | **Risk Response strategy** | **Contingency plan** | **Owner** | **Status** | **Date updated** |
| Sensitive data regarding patient and nurse details being leaked | **High**. Any vulnerabilities that results in exposing sensitive data will be detrimental to the hospital | High | **Avoid**. We can run everything locally, rather than on the web. | Backup and encrypt the data | Zach  Miguel | closed | Sept, 29 |
| Working out of scope by neglecting essential features | Wasting time on tasks that don’t lead to the client’s goals | Moderate | **Mitigate**. Always refer to project goals and deliverables to stay on track. | Refocus project goals and deliverables | All | closed | Sept, 29 |
| Poor coding | Applications might not be scalable and contain security flaws that ruin PR for the company | High | **Mitigate**. Follow good coding practices to reduce the risk impact as much as possible. | Debug the code, optimize, and/or reconstruct. | Development team | closed | Sept, 29 |
| Lack of communication | Lack of clarity and increased confusion. | Moderate | **Mitigate**. Always have weekly meetings and open communication channels | Connect with the client and set up communication channels and planned communication | All | closed | Sept, 29 |

**Recommendations**

One of the lessons learned during the development of this project was to let the clients try out each deliverable earlier in the process. This allows the client to get a feel for the application and provide effective feedback based on their experience. As our team was a bit late on this process, there happened to be cases of scope creep and desired features that were not able to be implemented on time. We recommend that teams working on this type of project should let the clients test out features earlier in the development process in order to identify new features that may need to be implemented.

Here are some directions for the remaining features to be added:

**Algorithm:** The current assignment algorithm does not provide a global optimum when picking the best nurse/patient assignment. Ideally, the algorithm should be able to iterate through all possible eligible nurses to provide the best fit for the assignments as a whole, instead of picking the first most fitting nurse for that patient. The algorithm also is coded based on a “golden goldilocks” situation, meaning the inputs are perfect and valid for the algorithm. This also means that the algorithm can’t handle edge cases, such as if there isn’t enough high skilled nurses to accommodate high acuity patients

**Database:** Currently the nurses use a scheduling program called HASS which outputs a CSV file. Ideally Smart Roster would be able to import that CSV file via a script so that nurses can be populated into Smart Roster automatically.

**Security:** The client would like to save past nurse/patient assignments for up to a year or more. In order to do this properly, sensitive information would must be saved and encrypted in a way that complies with their standards.

**Deployment:** The current application is designed to run on one local computer, as it uses a local MySQL database. However, the ideal goal was to deploy the application on multiple computers, which would require a database server maintained by the hospital’s IT team. In addition, all dependencies and packages needed to run the application would need to be installed on the client’s computers. It is important to verify that these computers are given permissions to install new programs, and recommended that these dependencies are installed by running a script.

**Usability:** They want to be able to see Nurse and Patient details in the Nurse/Patient Assignment Page. They want the ability to automatically feed a future shift lineup into the current shift when the time comes. They want to be able to click and drag nurses and patients.