**Team 21 Sprint II Retrospective**

**Project Name: STABLE**

**­**Adam Baker, Pedro Del Moral Lopez, Roy Ramstad, Shantanu Nair

***What Went Well?***

In Sprint 2, we were able to implement every user story from our planning document, as well as user authentication features. Since we were new to Calendar API, OAuth, and D3, much of our time was spent learning how to use them to produce the results we were looking for. We also needed to figure out how to make a custom search bar and pop-up modals for data entry. Once we overcame these challenges, our sprint became extremely productive, and we delivered more than we expected.

***Following are the tasks that were successfully completed in Sprint 2.***

1. **Training Regimen Database and User Interface**

* The user interface was successfully created and error checking was successfully implemented. Users were able to create a training regimen by filling out a form with all the information and sending it to the database. Then the profile would be updated and the user could view and edit all the information. This user story is complete.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Status** | **Owner** |
| 1 | Implement training regimen database | Completed | Shantanu |
| 2 | Connect the front end to the backend | Completed | Roy & Pedro |
| 3 | Implement a “database” page for the training regimen | Completed | Shantanu |
| 4 | Implement searching functionality for the training regimens | Completed | Roy |
| 5 | Implement error handling for invalid searches | Completed | Shantanu |
| 6 | Implement drop-down list of horses as user types | Completed | Roy |
| 7 | Implement clickable links on each of the items | Completed | Roy |

1. **Search Functionality for Horses, Vaccines, and Training Regimens**

* The search functionality on the main page is complete. Users can select whether they want to search for vaccines, training regimens, or horses, and see suggestions as they type. When they click on an option, they will be sent to the selected profile.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Status** | **Owner** |
| 1 | Implement searching functionality for the vaccines | Completed | Roy |
| 2 | Implement error handling for invalid searches | Completed | Roy |
| 3 | Implement drop-down list of vaccines as user types | Completed | Roy |
| 4 | Implement clickable links on each of the items | Completed | Roy |

1. **User Interface for Adding Vaccines to Horse Profiles**

* Vaccines can now be added to the horse profiles. When the user clicks “Add Vaccines,” they can see a list of previously added vaccines and add as many as they like to the horse profile. This information pops up in a modal on the same page. As vaccines are added, the user interface updates to display the new information.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Status** | **Owner** |
| 1 | Implement ability to search for existing vaccines while editing the horse form | Completed | Roy & Pedro |
| 2 | Implement ability to select multiple vaccines | Completed | Pedro |
| 3 | Implement ability to edit and remove vaccines from a particular horse | Completed | Pedro |
| 4 | Implement ability to add a new vaccine from a modal while editing the horse form | Completed | Pedro & Roy |
| 5 | Implement medical history information webpage | Completed | Roy & Pedro |

1. **User Interface for Adding Training Regimens to Horse Profiles**

* Training regimens can now be added to the horse profiles. When the user clicks “Add Training Programs,” they can see a list of previously added training regimens and add as many as they like to the horse profile, using a modal similar to the vaccines. As training regimens are added, the user interface updates to display the new information.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Status** | **Owner** |
| 1 | Implement ability to search for existing training regimens while editing the horse form | Completed | Roy |
| 2 | Implement ability to select multiple training regimens | Completed | Shantanu |
| 3 | Implement ability to edit and remove training regimens for a particular horse | Completed | Shantanu |
| 4 | Implement ability to add a new training regimen from a modal while editing the horse form | Completed | Pedro & Roy |

1. **User Interface for Importing and Exporting Horses, Vaccines, and Training Regimens**

* The user interface and backend functionality for importing and exporting was successfully implemented. The user can add horses, vaccines, and training regimens to a queue to be exported by clicking “Add to Export Queue.” When the user clicks “Export” to export that information, a file is created that contains the exported data. At that point, the data can be imported from another account by selecting “Import” and selecting the exported file.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Status** | **Owner** |
| 1 | Implement ability to add horse to a queue | Completed | Shantanu & Pedro |
| 2 | Implement ability to add vaccine to a queue | Completed | Shantanu & Pedro |
| 3 | Implement ability to add training regimen to a queue | Completed | Shantanu & Pedro |
| 4 | Implement ability to export queue | Completed | Shantanu |
| 5 | Implement ability to import horses | Completed | Shantanu |
| 6 | Implement ability to import vaccines | Completed | Shantanu |
| 7 | Implement ability to import training regimens | Completed | Shantanu |

1. **User Interface and Integration with Google Calendar API for Reminders**

* The user interface on the main page was updated to show the next three upcoming reminders that the user has set up from the calendar page. If the user is authenticated, the client sends a request to Google Calendar API for the user’s next three calendar events. When the user clicks “View All,” they can see the calendar with all their upcoming events. When an event is the next day, the user receives an email notification.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Status** | **Owner** |
| 1 | Implement integration with calendar API | Completed | Adam |
| 2 | Implement ability to set reminders | Completed | Adam |
| 3 | Implement ability to edit reminders | Completed | Adam |
| 4 | Implement ability to remove reminders | Completed | Adam |
| 5 | Implement ability to view top 3 upcoming reminders from the main page | Completed | Adam |
| 6 | Implement ability to send email reminders | Completed | Adam |

1. **User Interface and Integration with Google Calendar API for Traditional Calendar**

* The event creation and integration with Google Calendar API was successfully implemented. When the user navigates to the calendar page, they will see their personal calendar with the ability to edit, delete, and view more details by clicking on the events. When the user clicks “Add Event,” a modal pops up that allows them to enter event information. When the information is valid and the user clicks “OK,” the event will be sent to Google Calendar API and the calendar view will be updated to show the new information.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Status** | **Owner** |
| 1 | Implement integration with calendar API | Completed | Adam |
| 2 | Implement ability to view upcoming events month-by-month | Completed | Adam |
| 3 | Implement ability to click on a day and view all the events on that day | Completed | Adam |
| 4 | Implement ability to remove and edit calendar events | Completed | Adam |
| 5 | Implement ability to associate a set of calendar events with an individual horse if necessary | Completed | Adam |

1. **User Interface and Integration with Google Calendar API for Radial Calendar**

* The radial calendar was successfully implemented with the ability to view and click events, see the current day, and rotate the trimester wheel using a slider. The radial calendar was implemented using D3 to calculate the current day and segment the wheel into months and days. The events are pulled from Google Calendar API and added to the radial calendar on the correct dates. When the user clicks on a date with an event, they are taken to a page that shows them a list of all the events on that day.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Description** | **Status** | **Owner** |
| 1 | Implement radial calendar from scratch | Completed | Adam |
| 2 | Implement ability to set the start date based on the horse’s date of birth | Completed | Adam |
| 3 | Implement ability to point calendar to the current day | Completed | Adam |
| 4 | Implement ability to display important dates | Completed | Adam |
| 5 | Implement ability to view as a traditional calendar if preferred | Completed | Adam |

1. **User Authentication and Multiple Accounts**

* The user authentication and ability to use multiple accounts was successfully implemented. When the user visits the login page, they can login with their Google account and their authentication is carried around with a token. Every user with a separate account has their own separate set of horses, vaccines, events, etc.

***What Did Not Go Well?***

In Sprint 2, we did not have any unsuccessful user stories. The only difficulties we experienced were related to user authentication, because it caused us to have to rewrite most of our database calls. In regard to calendar API and D3, it took some trial and error to achieve our desired outcomes, but in the end, we found a way to implement every user story.

***Following are the tasks that were difficult or unsuccessful in Sprint 2.***

1. **Difficulties getting authorization to work**

* Our biggest difficulty during this sprint was figuring out how to get the authorization to work. It was a bit bumpy and we should’ve done it sooner, but we can improve next time by focusing on setting up farms faster. We had to change the code from one SDK to another, which took a long time, but it provided us with a great learning experience. Although this task proved to be harder than we anticipated, we ended up figuring it out and completing the user story successfully.

***How Should We Improve?***

* For Sprint 3, we will improve by being more thorough in our testing to make sure our application has as few bugs as possible.
* We plan to increase team collaboration for Sprint 3 because it will only be two weeks. We plan to maximize our time and get our user stories implemented faster than in previous sprints.
* We plan to stick more closely to our planning document in Sprint 3. In previous sprints, we sometimes deviated from the original plan. While this ultimately helped us out in the long run, it would be better if we made a reasonable plan from the start and stuck to it.
* With our experience from Sprint 2, we can also give better work estimates and allocate the proper amount of work to each team member. Just like after Sprint 1, we have more experience and can better predict how long each user story will take.