# KANM Show Scheduling - Sprint 3 Retrospective

**Software Engineering CSCE 606** **November 8th**

**Product Owner** – Haridher Pandiyan

**Scrum Master** – James Nojek

**Developers** – Ali Nablan, Ankit Mohanty, Davis Beilue, James Nojek, Kriti Sarker, Neeraj Julian Joseph Rajkumar, Toan Vu, Haridher Pandiyan

**Pivotal** - https://www.pivotaltracker.com/n/projects/2721031

**GitHub** - https://github.com/amohanty03/KANM-Show-Scheduling

**Deployed App** - https://kanm-show-scheduler-b962465e9890.herokuapp.com/

**Dates -** October 19, 2024 to November 1, 2024

| Member | Role | Points |
|---|---|---|
| Ali Nablan | Developed the third stage of the scheduling algorithm. | 3 |
| Ankit Mohanty | Developed the spreadsheet parser and improved the UI. | 3 |
| Davis Beilue | Developed the third stage of the scheduling algorithm. | 3 |
| James Nojek (Scrum Master) | Developed the spreadsheet parser and the calendar page and its functionality. | 4 |
| Kriti Sarker | Developed the first stage of the scheduling algorithm. | 2 |
| Neeraj Julian Joseph Rajkumar | Developed the first stage of the scheduling algorithm. | 2 |
| Toan Vu | Developed the second stage of the scheduling algorithm. | 1 |
| Haridher Pandiyan (Product Owner) | Developed the second stage of the scheduling algorithm. | 1 |

## Sprint Goal

The goal of this sprint is to complete the generate schedule function and improve the UI further. The function will be complete when a user can select an uploaded spreadsheet, click the generate button that generates the required schedule. The user will then be redirected to a page where the eventual schedule will be displayed.

## Sprint Achievements

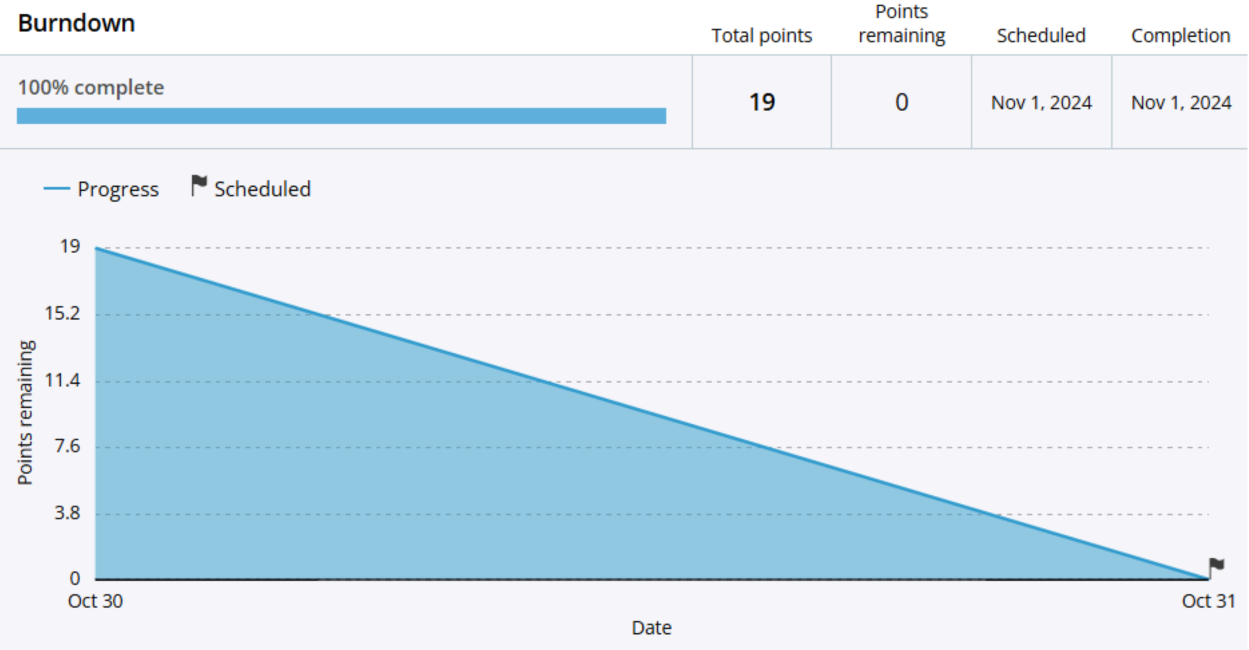
The sprint goals were achieved. The generate schedule algorithm has been developed and is assigning slots to the RJs based on the criterias defined by the stakeholders. The UI has had further improvement and looks a lot more appealing. Coverage in both cucumber and rspec continues to remain at over 90 % and the code climate remains at A. Stakeholders were happy with the progress made.

## Sprint Backlog

The features that were set out at the start of the sprint were mostly covered, but some extra cases that would be needed to cover the complete feature completely were spilled over and moved to the other sprint to keep things within limits in this sprint while maintaining code coverage. Some stories were changed up in their implementation during the course of the sprint through good ideas that were shared by the team.

| Story | Points | Reason |
|---|---|---|
| - Update the Schedule Algorithm to handle Co-hosts. | 1 | This scenario was noticed during our testing within the sprint and hence had to be kept for the next sprint. |
| * Rank the Radio Jockeys | 2 | A better implementation was identified for this, and hence covered. |
| - Perform a better mapping of the spreadsheet to the RJ table. | 1 | Needed more time to finalize the form changes without blocking the team. |

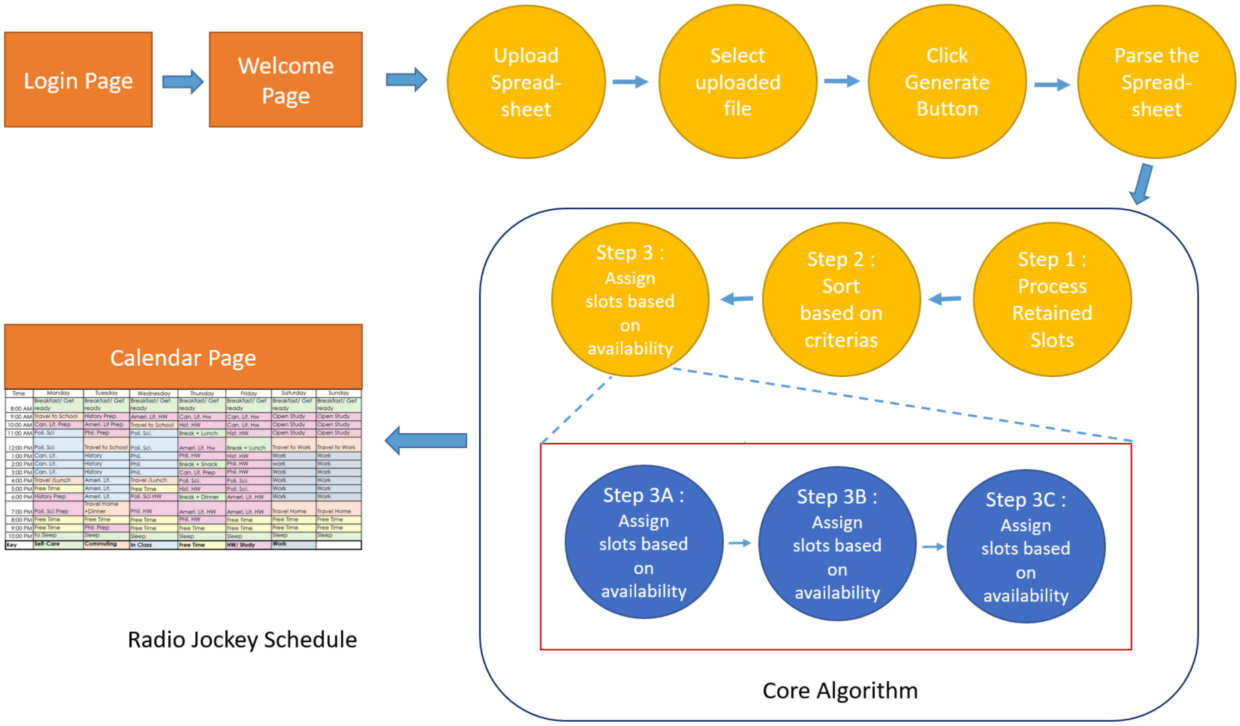
## Burndown Chart



Burndown

| 100% complete | Total points | Points remaining | Scheduled | Completion |
|---|---|---|---|---|
| | 19 | 0 | Nov 1, 2024 | Nov 1, 2024 |

## Design Diagrams



Login Page
Welcome Page
Upload Spread-sheet
Select uploaded file
Click Generate Button
Parse the Spread-sheet
Step 1 : Process Retained Slots
Step 2 : Sort based on criterias
Step 3 : Assign slots based on availability
Step 3A : Assign slots based on availability
Step 3B : Assign slots based on availability
Step 3C : Assign slots based on availability
Core Algorithm
Calendar Page
Radio Jockey Schedule

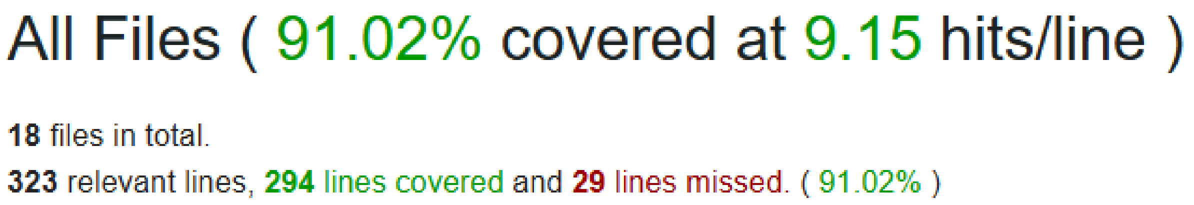
## Documentation Of Changes

This sprint we began to work on the core algorithm of our application and its implementation in parsing the submitted spreadsheet and generating the required schedule when the Generate button is clicked. The results are currently displayed without any CSS formatting, so that will be worked on in the following sprint.

Stories, mockups and storyboards are documented as part of the Sprint 3 document.
https://github.com/amohanty03/KANM-Show-Scheduling/blob/main/documentation/Fall2024/Sprint_3.pdf

## Evaluation Of Code And Test Quality

More than 90% of the code has been tested, which is evident from the code coverage report :

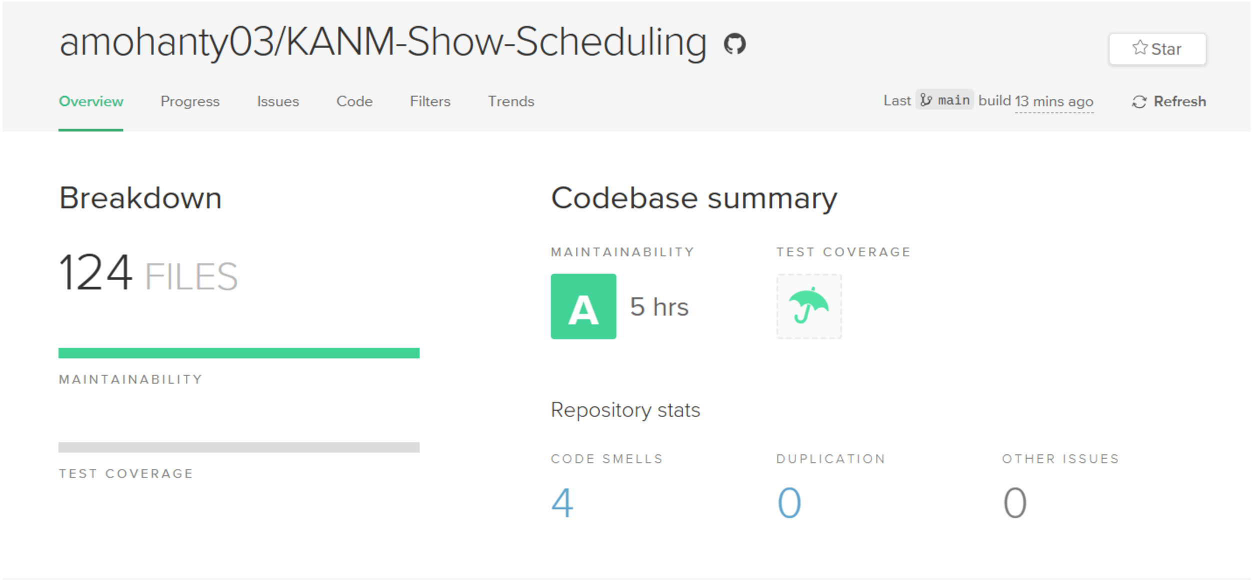


All Files ( 91.02% covered at 9.15 hits/line )

18 files in total.

323 relevant lines, 294 lines covered and 29 lines missed. ( 91.02% )

Rubocop was executed and no offenses were detected.
Code climate report has given a rating ‘A’ with 4 code smells.
Code Climate Report https://codeclimate.com/github/amohanty03/KANM-Show-Scheduling



amohanty03/KANM-Show-Scheduling

Breakdown

124 FILES

MAINTAINABILITY

TEST COVERAGE

Codebase summary

MAINTAINABILITY

A 5 hrs

TEST COVERAGE

Repository stats

| CODE SMELLS | DUPLICATION | OTHER ISSUES |
|---|---|---|
| 4 | 0 | 0 |

## Customer Meeting

October 31$^{th}$ 2024, 6:30 pm
Virtual Meet

The form changes were presented to the Client, which they agreed to. We've requested them to integrate it into theirs and return the final version. We will provide the form shortly. They are also fine with just doing the xlsx file upload.

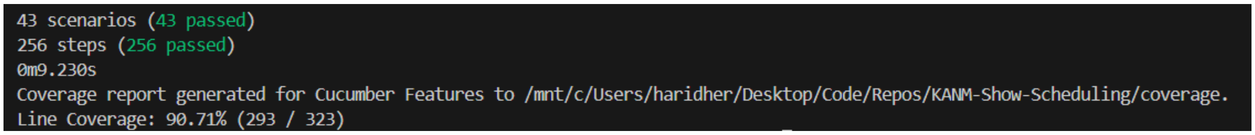
We presented a demo of the current progress which they are satisfied with.

For the co-host, they told us to go with the RJ member with higher priority, so if 2 members have different seniority, go with the data of the more senior member.

## BDD & TDD

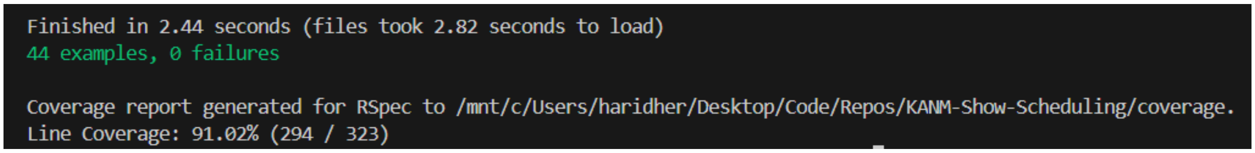
Cucumber Scenarios and Rspec examples have been written in such a way that they cover the entirety of the changes made to the welcome page to account for the form changes while handling delete and generate action based on selected files, and ofcourse coverage of the main feature developed this sprint, which is the scheduling algorithm itself.

43 Cucumber scenarios



```
43 scenarios (43 passed)
256 steps (256 passed)
0m9.230s
Coverage report generated for Cucumber Features to /mnt/c/Users/haridher/Desktop/Code/Repos/KANM-Show-Scheduling/coverage.
Line Coverage: 90.71% (293 / 323)
```

44 Rspec examples



```
Finished in 2.44 seconds (files took 2.82 seconds to load)
44 examples, 0 failures

Coverage report generated for RSpec to /mnt/c/Users/haridher/Desktop/Code/Repos/KANM-Show-Scheduling/coverage.
Line Coverage: 91.02% (294 / 323)
```