# Final Report and Documentation - EA Block Scheduling

## 1. Two paragraph summaries of the project as implemented.

This project addresses the need for an efficient and automated solution to create and manage block schedules for math, science, and engineering classes offered by Texas A&M University (TAMU) and its community college partner. Currently, block scheduling is a manual process involving spreadsheets, calendars, and Google Forms, which is time-consuming and prone to errors. To streamline this process, the application allows admins, including college staff, professors, and administrators, to upload class schedules in Excel format. The uploaded data is parsed into a database and serves as input for a scheduling algorithm that generates non-overlapping class blocks while considering prerequisites and capacity constraints. Each block contains at least one class from each subject area to meet academic requirements. The algorithm generates all possible valid combinations, ensuring flexibility and completeness, with the current input producing 67 unique blocks. Additionally, the 67 generated blocks can be exported into an Excel format, which may be needed for office use.

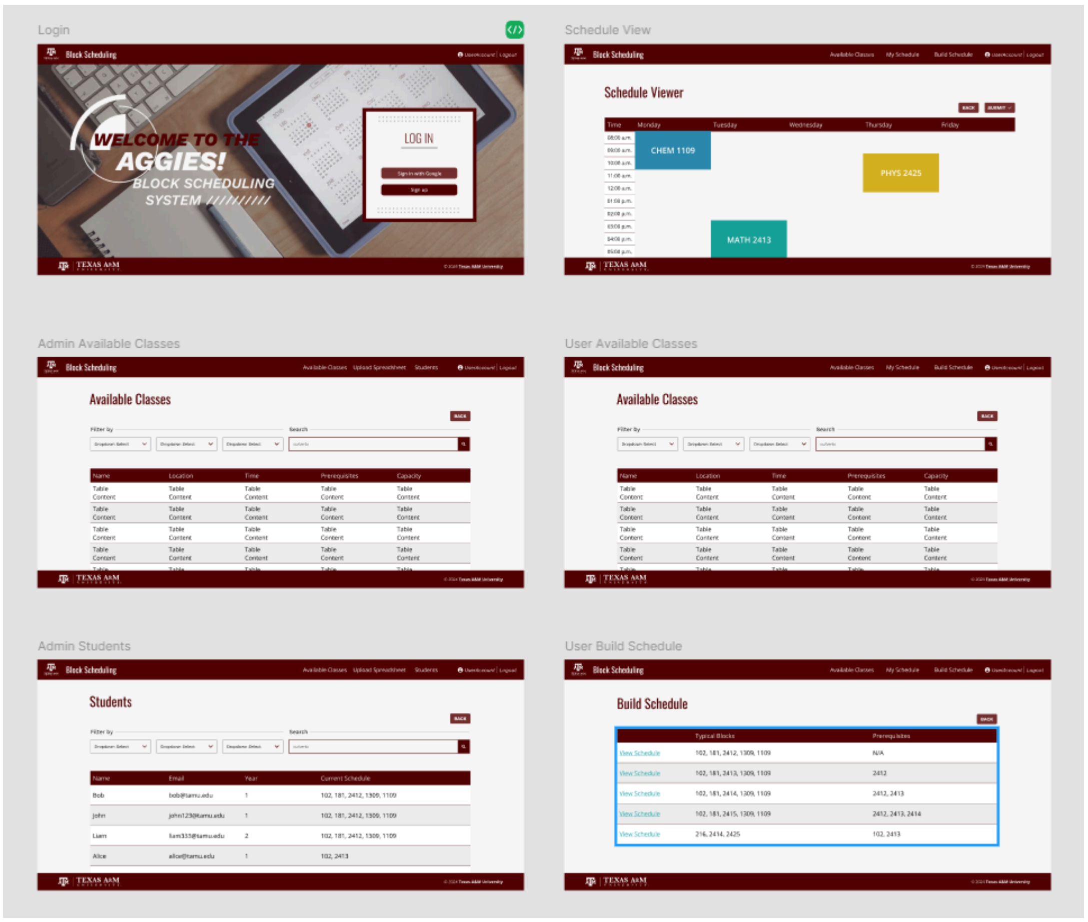
The application is designed to serve multiple stakeholders, including admins and students. Admins can upload class data and manage schedules, while students can explore available block combinations to plan their classes effectively. The system ensures security through Google account-based login, offering personalized profile pages for all users. The visually intuitive and spreadsheet-compatible output provides users with a clear and detailed view of class block options. By automating the block scheduling process and incorporating key constraints like prerequisites and non-overlapping timings, the application significantly reduces manual effort and enhances the scheduling experience for stakeholders.

## 2. Description of all user stories

1. **Sprint1 Documentations - 3 points**
    - As a developer,
      When I go to /documentation/Fall2024/ under the Github repository,
      I can see the documentation for sprint 1.
    - Sprint 1
    - Status: Completed

2. **UI Mock-up – 8 points**
    - As a developer,
      When I go to /assets/Pages/ under the Github repository,
      I can see the UI mock-ups,
    - Then I go to assets/Widgets,
      I can see the UI widgets.
    - Sprint 1
    - Status: Completed

3. **System structure diagram – 2 points**
    - As a developer,
      When I go to /documentation/Fall2024/ under the Github repository,
    - Then I open one of the sprint plans,
      I can see the system structure diagram
    - Sprint 1
    - Status: Completed



**Available Classes**

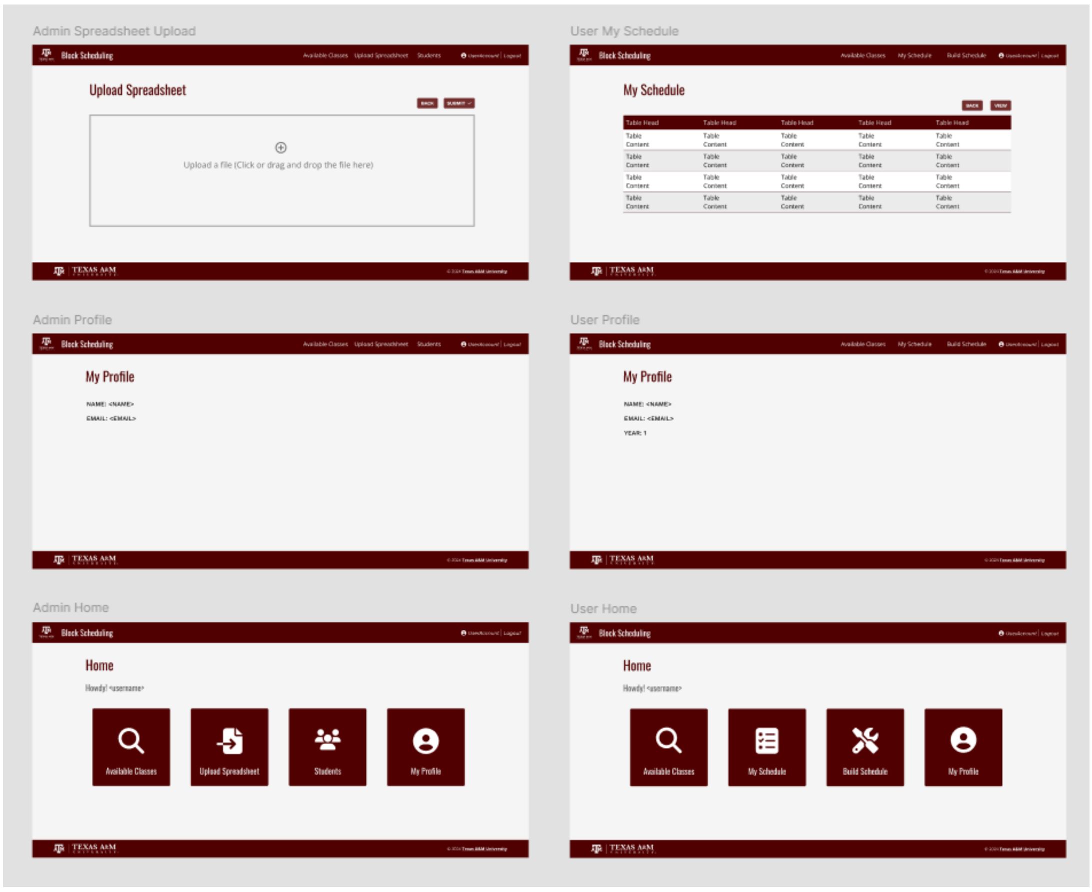
| Name | Location | Time | Prerequisites | Capacity |
|---|---|---|---|---|
| Table Content | Table Content | Table Content | Table Content | Table Content |
| Table Content | Table Content | Table Content | Table Content | Table Content |
| Table Content | Table Content | Table Content | Table Content | Table Content |
| Table Content | Table Content | Table Content | Table Content | Table Content |

**Students**

| Name | Email | Year | Current Schedule |
|---|---|---|---|
| Bob | bob@tamu.edu | 1 | 102, 181, 2412, 1309, 1109 |
| John | john123@tamu.edu | 1 | 102, 181, 2412, 1309, 1109 |
| Liam | liam333@tamu.edu | 2 | 102, 181, 2412, 1309, 1109 |
| Alice | alice@tamu.edu | 1 | 102, 2413 |

**Build Schedule**

| | Typical Blocks | Prerequisites |
|---|---|---|
| View Schedule | 102, 181, 2412, 1309, 1109 | N/A |
| View Schedule | 102, 181, 2413, 1309, 1109 | 2412 |
| View Schedule | 102, 181, 2414, 1309, 1109 | 2412, 2413 |
| View Schedule | 102, 181, 2415, 1309, 1109 | 2412, 2413, 2414 |
| View Schedule | 216, 2414, 2425 | 102, 2413 |



**My Schedule**

| Table Head | Table Head | Table Head | Table Head | Table Head |
|---|---|---|---|---|
| Table Content | Table Content | Table Content | Table Content | Table Content |
| Table Content | Table Content | Table Content | Table Content | Table Content |
| Table Content | Table Content | Table Content | Table Content | Table Content |
| Table Content | Table Content | Table Content | Table Content | Table Content |

**My Profile**

NAME: <NAME>
EMAIL: <EMAIL>

**My Profile**

NAME: <NAME>
EMAIL: <EMAIL>
YEAR: 1

4. **Sprint2 Documentations - 3 points**
    - As a developer,
      When I go to /documentation/Fall2024/ under the Github repository,
      I can see the documentation for sprint 2.
    - Sprint 2
    - Status: Completed
5. **Display Class Blocks - 5 points**
    - As a user,
    - I open the schedule viewer page,
    - I can see the timetable,
    - And I can see the class blocks displayed correctly
    - Sprint 3
    - Status: Completed
6. **Display Class Details - 5 points**
    - As a user,
    - When I open the schedule viewer page,
    - I can see the class blocks,
    - Then I hover my mouse on the class block,
    - I can see the details of this class ( Instructor, Location...)
    - Sprint 3
    - Status: Completed
7. **Show basic schedule timetable - 3 points**
    - As a user
      I want to see my schedule chart
      Then I open the schedule viewer page
      And I see a chart with time blocks.
    - Sprint 3
    - Status: Completed
8. **Gemfile fix – 2 points**
    - As a developer,
      When I set up the environment,
      I shouldn't see duplicate settings in the gemfile
    - Sprint 4
    - Status: Completed
9. **Make each classes object - 13 points**
    - As a developer,
    - I add class objects when I generate the schedule. Then I see two possible schedules.
    - Sprint 1
    - Status: Completed
10. **Set typical classes grouping on the web - 5 points**
    - As an admin,
    - admin sets typical classes to generate draft blocks on the web page typical blocks are generated
    - Sprint 2,3
    - Status: Completed

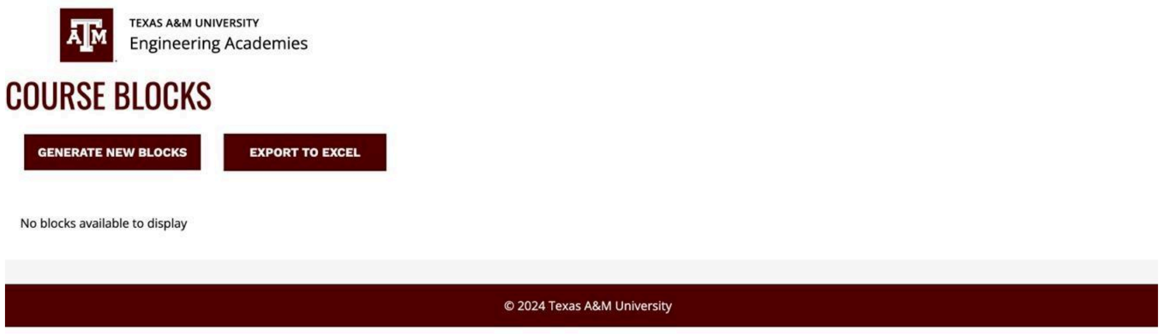
11. **Set block generating conditions on the web - 10 points**
    - As an admin,
    - admin sets block generating conditions including prerequisite and conditions and block generating conditions are made in web application
    - Sprint 2,3
    - Status: Completed

12. **Application makes possible blocks with conditions - 8 points**
    - As a developer,
    - I make an algorithm to apply block generating conditions for typical blocks made by admin. Blocks are added to typical blocks
    - Sprint 2,3
    - Status: Completed

13. **Generate possible blocks - 8 points**
    - As a developer,
    - I make code that makes possible blocks automatically. Blocks are generated automatically through algorithm
    - Sprint 4
    - Status: Completed

14. **User pick rest of classes - 5 points**
    - As a user,
    - user choose a block and if the chosen block is not possible for the user, pop up alerts user that "scheduled blocks limited!". When users pick an unavailable block, there will be a warning sign popping up on the web
    - Sprint 4
    - Status: Completed

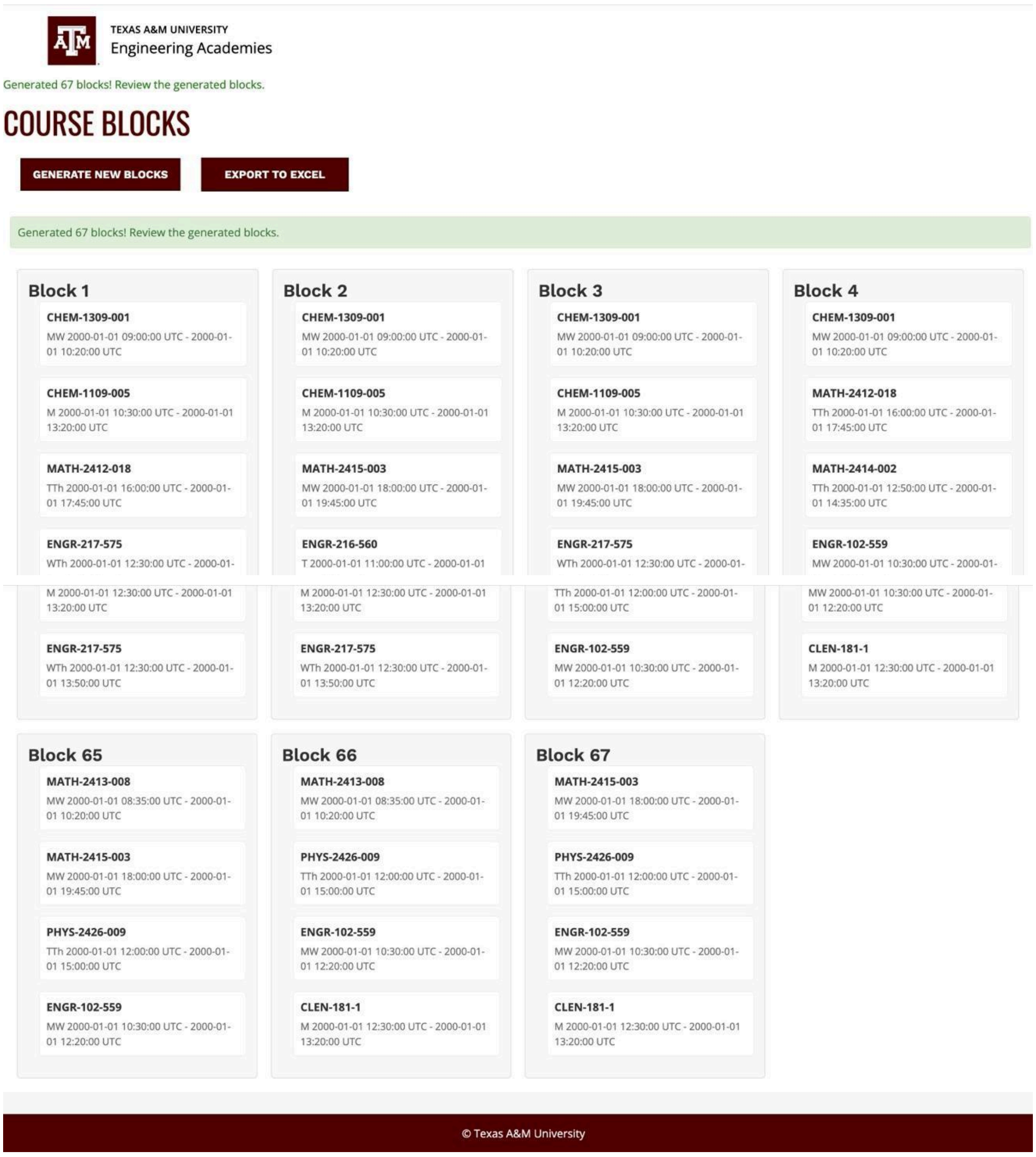
1. Course blocks



No blocks available to display

© 2024 Texas A&M University

2. Block Generation



TEXAS A&M UNIVERSITY
Engineering Academies

Generated 67 blocks! Review the generated blocks.

COURSE BLOCKS

GENERATE NEW BLOCKS · EXPORT TO EXCEL

Generated 67 blocks! Review the generated blocks.

Block 1

CHEM-1309-001
MW 2000-01-01 09:00:00 UTC - 2000-01-01 10:20:00 UTC

CHEM-1109-005
M 2000-01-01 10:30:00 UTC - 2000-01-01 13:20:00 UTC

MATH-2412-018
TTh 2000-01-01 16:00:00 UTC - 2000-01-01 17:45:00 UTC

ENGR-217-575
WTh 2000-01-01 12:30:00 UTC - 2000-01-

Block 2

CHEM-1309-001
MW 2000-01-01 09:00:00 UTC - 2000-01-01 10:20:00 UTC

CHEM-1109-005
M 2000-01-01 10:30:00 UTC - 2000-01-01 13:20:00 UTC

MATH-2415-003
MW 2000-01-01 18:00:00 UTC - 2000-01-01 19:45:00 UTC

ENGR-216-560
T 2000-01-01 11:00:00 UTC - 2000-01-01

Block 3

CHEM-1309-001
MW 2000-01-01 09:00:00 UTC - 2000-01-01 10:20:00 UTC

CHEM-1109-005
M 2000-01-01 10:30:00 UTC - 2000-01-01 13:20:00 UTC

MATH-2415-003
MW 2000-01-01 18:00:00 UTC - 2000-01-01 19:45:00 UTC

ENGR-217-575
WTh 2000-01-01 12:30:00 UTC - 2000-01-

Block 4

CHEM-1309-001
MW 2000-01-01 09:00:00 UTC - 2000-01-01 10:20:00 UTC

MATH-2412-018
TTh 2000-01-01 16:00:00 UTC - 2000-01-01 17:45:00 UTC

MATH-2414-002
TTh 2000-01-01 12:50:00 UTC - 2000-01-01 14:35:00 UTC

ENGR-102-559
MW 2000-01-01 10:30:00 UTC - 2000-01-

M 2000-01-01 12:30:00 UTC - 2000-01-01 13:20:00 UTC

ENGR-217-575
WTh 2000-01-01 12:30:00 UTC - 2000-01-01 13:50:00 UTC

M 2000-01-01 12:30:00 UTC - 2000-01-01 13:20:00 UTC

ENGR-217-575
WTh 2000-01-01 12:30:00 UTC - 2000-01-01 13:50:00 UTC

TTh 2000-01-01 12:00:00 UTC - 2000-01-01 15:00:00 UTC

ENGR-102-559
MW 2000-01-01 10:30:00 UTC - 2000-01-01 12:20:00 UTC

MW 2000-01-01 10:30:00 UTC - 2000-01-01 12:20:00 UTC

CLEN-181-1
M 2000-01-01 12:30:00 UTC - 2000-01-01 13:20:00 UTC

Block 65

MATH-2413-008
MW 2000-01-01 08:35:00 UTC - 2000-01-01 10:20:00 UTC

MATH-2415-003
MW 2000-01-01 18:00:00 UTC - 2000-01-01 19:45:00 UTC

PHYS-2426-009
TTh 2000-01-01 12:00:00 UTC - 2000-01-01 15:00:00 UTC

ENGR-102-559
MW 2000-01-01 10:30:00 UTC - 2000-01-01 12:20:00 UTC

Block 66

MATH-2413-008
MW 2000-01-01 08:35:00 UTC - 2000-01-01 10:20:00 UTC

PHYS-2426-009
TTh 2000-01-01 12:00:00 UTC - 2000-01-01 15:00:00 UTC

ENGR-102-559
MW 2000-01-01 10:30:00 UTC - 2000-01-01 12:20:00 UTC

CLEN-181-1
M 2000-01-01 12:30:00 UTC - 2000-01-01 13:20:00 UTC

Block 67

MATH-2415-003
MW 2000-01-01 18:00:00 UTC - 2000-01-01 19:45:00 UTC

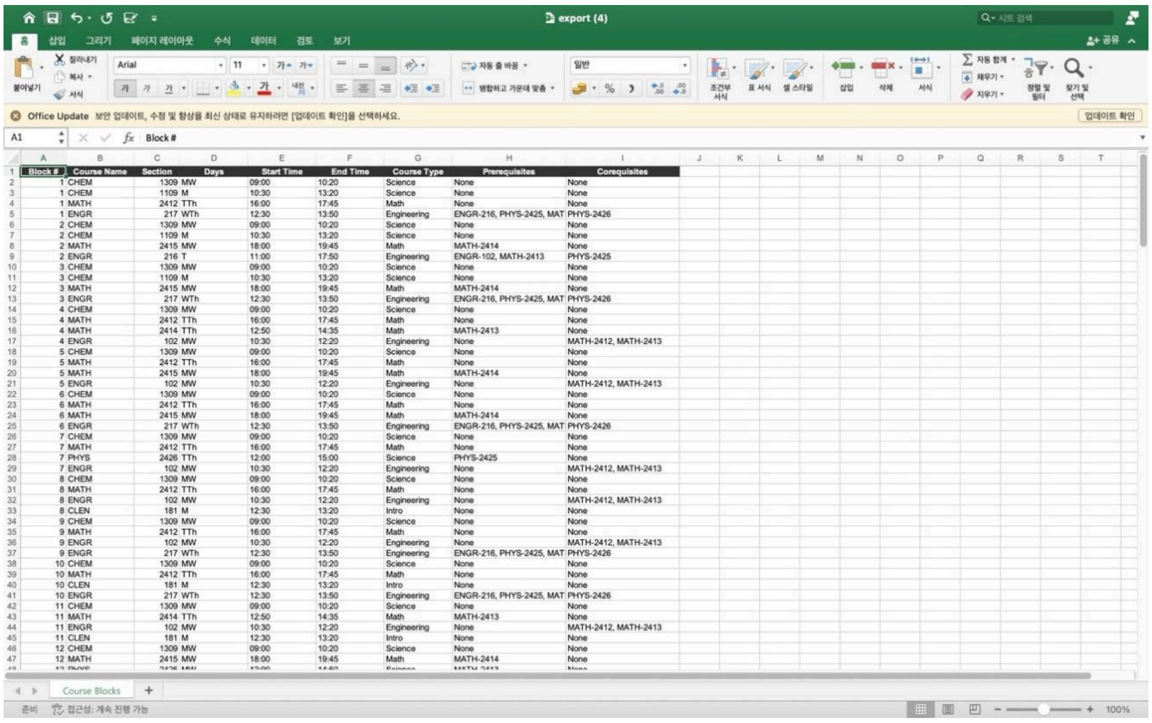
PHYS-2426-009
TTh 2000-01-01 12:00:00 UTC - 2000-01-01 15:00:00 UTC

ENGR-102-559
MW 2000-01-01 10:30:00 UTC - 2000-01-01 12:20:00 UTC

CLEN-181-1
M 2000-01-01 12:30:00 UTC - 2000-01-01 13:20:00 UTC

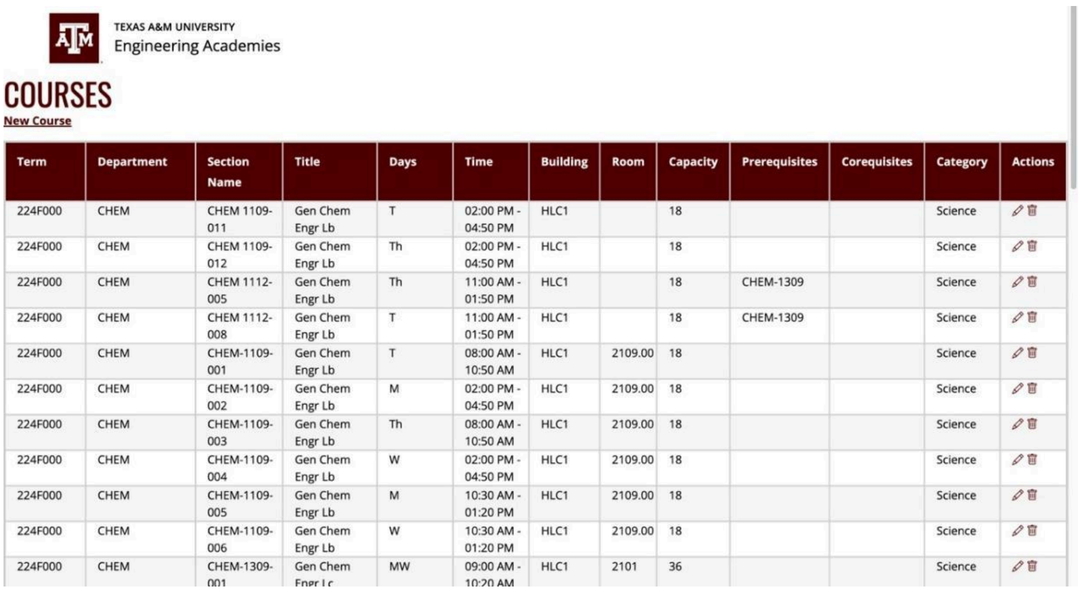
© Texas A&M University

3. Block schedule export to spreadsheet



| Block # | Course Name | Section | Days | Start Time | End Time | Course Type | Prerequisites | Corequisites |
|---|---|---|---|---|---|---|---|---|
| 1 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 1 | CHEM | 1109 | M | 10:30 | 13:20 | Science | None | None |
| 1 | MATH | 2412 | TTh | 16:00 | 17:45 | Math | None | None |
| 1 | ENGR | 217 | WTh | 12:30 | 13:50 | Engineering | ENGR-216, PHYS-2425, MAT | PHYS-2426 |
| 2 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 2 | CHEM | 1109 | M | 10:30 | 13:20 | Science | None | None |
| 2 | MATH | 2415 | MW | 18:00 | 19:45 | Math | MATH-2414 | None |
| 2 | ENGR | 216 | T | 11:00 | 17:50 | Engineering | ENGR-102, MATH-2413 | PHYS-2425 |
| 3 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 3 | CHEM | 1109 | M | 10:30 | 13:20 | Science | None | None |
| 3 | MATH | 2415 | MW | 18:00 | 19:45 | Math | MATH-2414 | None |
| 3 | ENGR | 217 | WTh | 12:30 | 13:50 | Engineering | ENGR-216, PHYS-2425, MAT | PHYS-2426 |
| 4 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 4 | MATH | 2412 | TTh | 16:00 | 17:45 | Math | None | None |
| 4 | MATH | 2414 | TTh | 12:50 | 14:35 | Math | MATH-2413 | None |
| 4 | ENGR | 102 | MW | 10:30 | 12:20 | Engineering | None | MATH-2412, MATH-2413 |
| 5 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 5 | MATH | 2412 | TTh | 16:00 | 17:45 | Math | None | None |
| 5 | MATH | 2415 | MW | 18:00 | 19:45 | Math | MATH-2414 | None |
| 5 | ENGR | 102 | MW | 10:30 | 12:20 | Engineering | None | MATH-2412, MATH-2413 |
| 6 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 6 | MATH | 2412 | TTh | 16:00 | 17:45 | Math | None | None |
| 6 | MATH | 2415 | MW | 18:00 | 19:45 | Math | MATH-2414 | None |
| 6 | ENGR | 217 | WTh | 12:30 | 13:50 | Engineering | ENGR-216, PHYS-2425, MAT | PHYS-2426 |
| 7 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 7 | MATH | 2412 | TTh | 16:00 | 17:45 | Math | None | None |
| 7 | PHYS | 2426 | TTh | 12:00 | 15:00 | Science | PHYS-2425 | None |
| 7 | ENGR | 102 | MW | 10:30 | 12:20 | Engineering | None | MATH-2412, MATH-2413 |
| 8 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 8 | MATH | 2412 | TTh | 16:00 | 17:45 | Math | None | None |
| 8 | ENGR | 102 | MW | 10:30 | 12:20 | Engineering | None | MATH-2412, MATH-2413 |
| 8 | CLEN | 181 | M | 12:30 | 13:20 | Intro | None | None |
| 9 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 9 | MATH | 2412 | TTh | 16:00 | 17:45 | Math | None | None |
| 9 | ENGR | 102 | MW | 10:30 | 12:20 | Engineering | None | MATH-2412, MATH-2413 |
| 9 | ENGR | 217 | WTh | 12:30 | 13:50 | Engineering | ENGR-216, PHYS-2425, MAT | PHYS-2426 |
| 10 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 10 | MATH | 2412 | TTh | 16:00 | 17:45 | Math | None | None |
| 10 | CLEN | 181 | M | 12:30 | 13:20 | Intro | None | None |
| 10 | ENGR | 217 | WTh | 12:30 | 13:50 | Engineering | ENGR-216, PHYS-2425, MAT | PHYS-2426 |
| 11 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 11 | MATH | 2414 | TTh | 12:50 | 14:35 | Math | MATH-2413 | None |
| 11 | ENGR | 102 | MW | 10:30 | 12:20 | Engineering | None | MATH-2412, MATH-2413 |
| 11 | CLEN | 181 | M | 12:30 | 13:20 | Intro | None | None |
| 12 | CHEM | 1309 | MW | 09:00 | 10:20 | Science | None | None |
| 12 | MATH | 2415 | MW | 18:00 | 19:45 | Math | MATH-2414 | None |

4. Course showing on the web



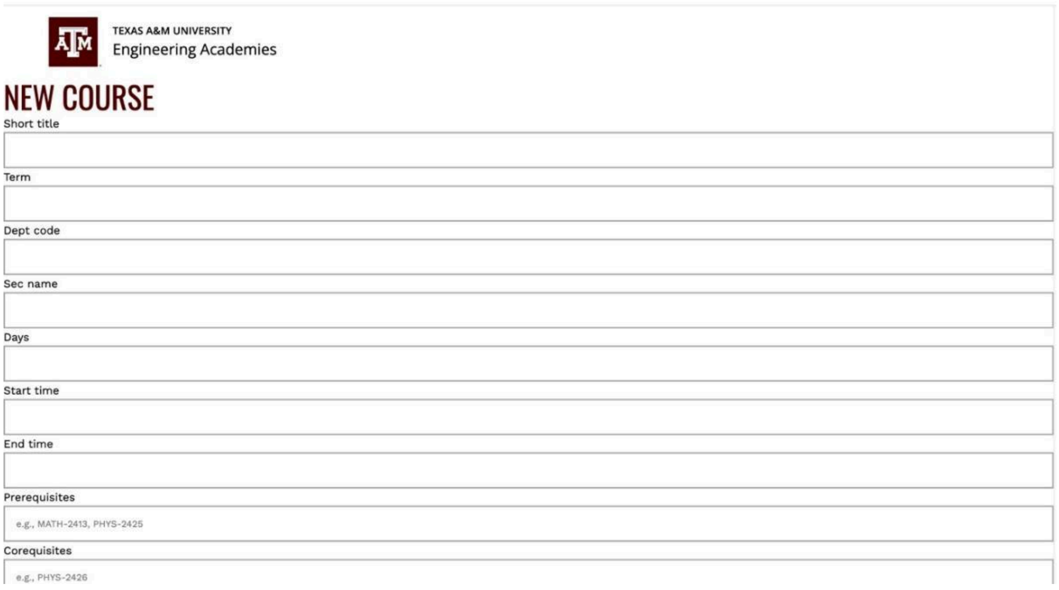
TEXAS A&M UNIVERSITY
Engineering Academies

**COURSES**

New Course

| Term | Department | Section Name | Title | Days | Time | Building | Room | Capacity | Prerequisites | Corequisites | Category | Actions |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 224F000 | CHEM | CHEM 1109-011 | Gen Chem Engr Lb | T | 02:00 PM - 04:50 PM | HLC1 | | 18 | | | Science | |
| 224F000 | CHEM | CHEM 1109-012 | Gen Chem Engr Lb | Th | 02:00 PM - 04:50 PM | HLC1 | | 18 | | | Science | |
| 224F000 | CHEM | CHEM 1112-005 | Gen Chem Engr Lb | Th | 11:00 AM - 01:50 PM | HLC1 | | 18 | CHEM-1309 | | Science | |
| 224F000 | CHEM | CHEM 1112-008 | Gen Chem Engr Lb | T | 11:00 AM - 01:50 PM | HLC1 | | 18 | CHEM-1309 | | Science | |
| 224F000 | CHEM | CHEM-1109-001 | Gen Chem Engr Lb | T | 08:00 AM - 10:50 AM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-002 | Gen Chem Engr Lb | M | 02:00 PM - 04:50 PM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-003 | Gen Chem Engr Lb | Th | 08:00 AM - 10:50 AM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-004 | Gen Chem Engr Lb | W | 02:00 PM - 04:50 PM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-005 | Gen Chem Engr Lb | M | 10:30 AM - 01:20 PM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-006 | Gen Chem Engr Lb | W | 10:30 AM - 01:20 PM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1309-001 | Gen Chem Engr Lc | MW | 09:00 AM - 10:20 AM | HLC1 | 2101 | 36 | | | Science | |

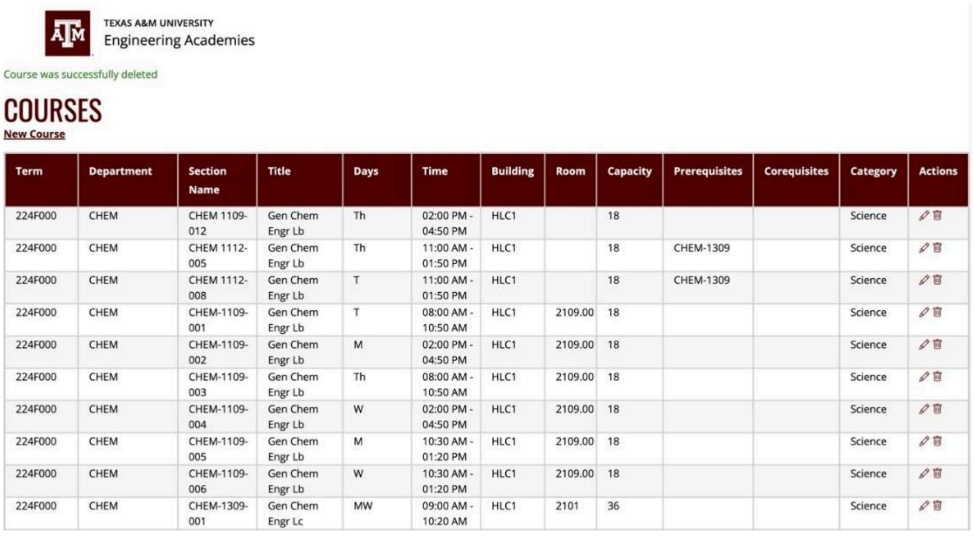
5. Course creation


TEXAS A&M UNIVERSITY
Engineering Academies

NEW COURSE

| | |
|---|---|
| Short title | |
| Term | |
| Dept code | |
| Sec name | |
| Days | |
| Start time | |
| End time | |
| Prerequisites | e.g., MATH-2413, PHYS-2425 |
| Corequisites | e.g., PHYS-2426 |

6. Course deletion


TEXAS A&M UNIVERSITY
Engineering Academies

Course was successfully deleted

COURSES

New Course

| Term | Department | Section Name | Title | Days | Time | Building | Room | Capacity | Prerequisites | Corequisites | Category | Actions |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 224F000 | CHEM | CHEM 1109-012 | Gen Chem Engr Lb | Th | 02:00 PM - 04:50 PM | HLC1 | | 18 | | | Science | |
| 224F000 | CHEM | CHEM 1112-005 | Gen Chem Engr Lb | Th | 11:00 AM - 01:50 PM | HLC1 | | 18 | CHEM-1309 | | Science | |
| 224F000 | CHEM | CHEM 1112-008 | Gen Chem Engr Lb | T | 11:00 AM - 01:50 PM | HLC1 | | 18 | CHEM-1309 | | Science | |
| 224F000 | CHEM | CHEM-1109-001 | Gen Chem Engr Lb | T | 08:00 AM - 10:50 AM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-002 | Gen Chem Engr Lb | M | 02:00 PM - 04:50 PM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-003 | Gen Chem Engr Lb | Th | 08:00 AM - 10:50 AM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-004 | Gen Chem Engr Lb | W | 02:00 PM - 04:50 PM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-005 | Gen Chem Engr Lb | M | 10:30 AM - 01:20 PM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1109-006 | Gen Chem Engr Lb | W | 10:30 AM - 01:20 PM | HLC1 | 2109.00 | 18 | | | Science | |
| 224F000 | CHEM | CHEM-1309-001 | Gen Chem Engr Lc | MW | 09:00 AM - 10:20 AM | HLC1 | 2101 | 36 | | | Science | |

**15. Upload the excel spreadsheet – 10 points**

- As an admin, when I open the spreadsheet upload page, I should see the upload button and then I press the button to upload file from local and I see the file has been successfully uploaded.

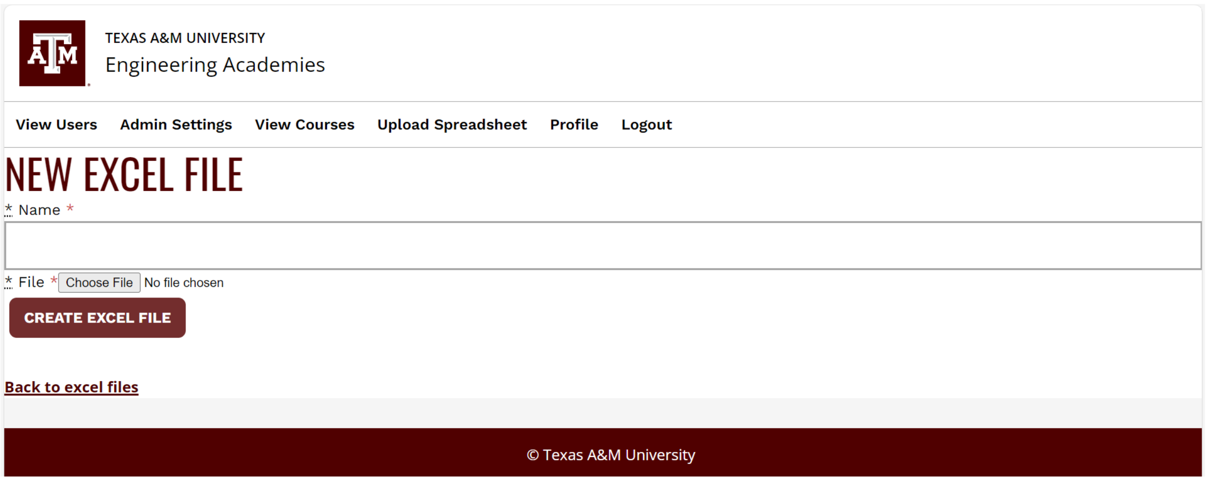
- Create page for uploading class spreadsheet
- Create parsing service for backend data persistence
- Add styling to upload page
- After uploading excel, it should be saved to the database
- Download the excel once uploaded and saved to database
- Destroy or delete the excel from the list and database
- Show the excel name and downloadable link and destroy excel option
- Integrate tasks 56 and 57
- Sprint 1,2
- Status: Completed

## 16. Issue fixed with the coverage of excel feature – 3 points

- As a developer, I should see test coverage > 90%
- Sprint 2
- Status: Completed

## 17. Store the spreadsheet data on the database – 20 points

- As a user after uploading the excel, the data should be populated on database table which can be viewed.
- Sprint 2,3
- Status: Completed



NEW EXCEL FILE

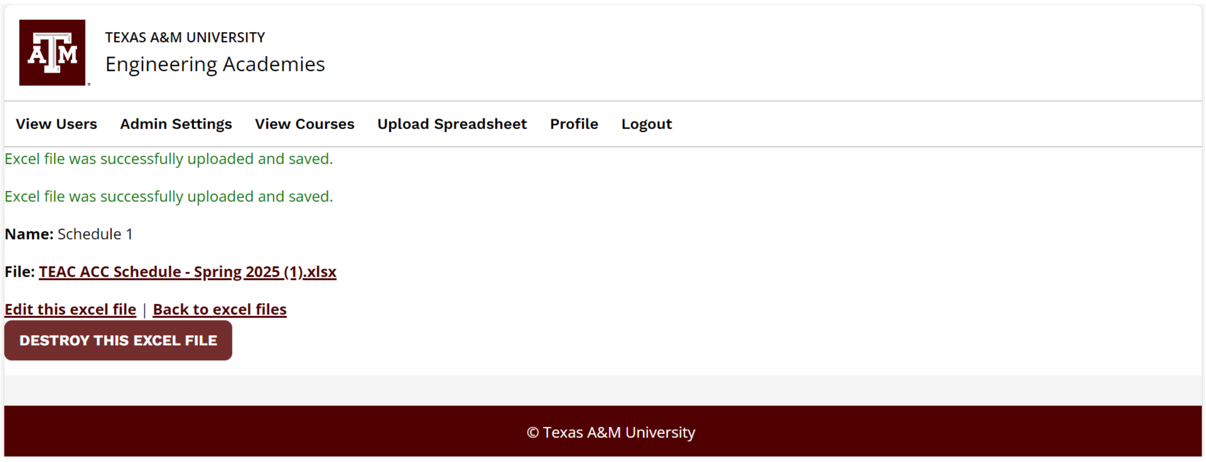
* Name *

* File * Choose File No file chosen

CREATE EXCEL FILE

Back to excel files

© Texas A&M University



Excel file was successfully uploaded and saved.

Excel file was successfully uploaded and saved.

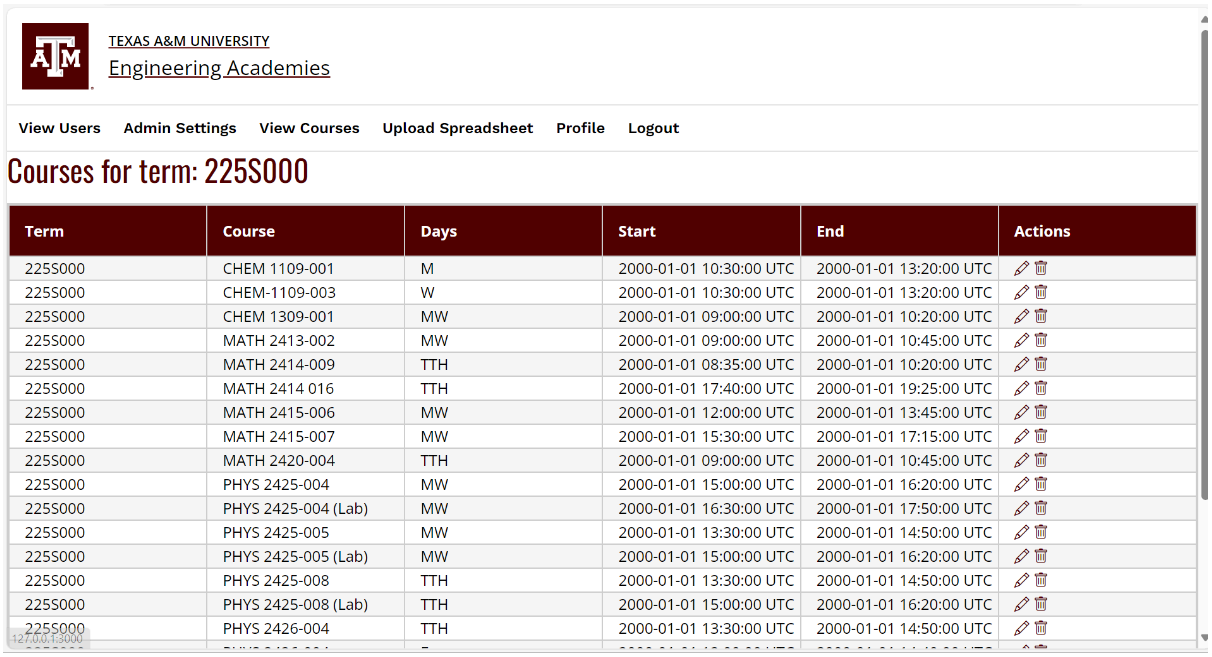
**Name:** Schedule 1

**File:** TEAC ACC Schedule - Spring 2025 (1).xlsx

Edit this excel file | Back to excel files

DESTROY THIS EXCEL FILE

© Texas A&M University

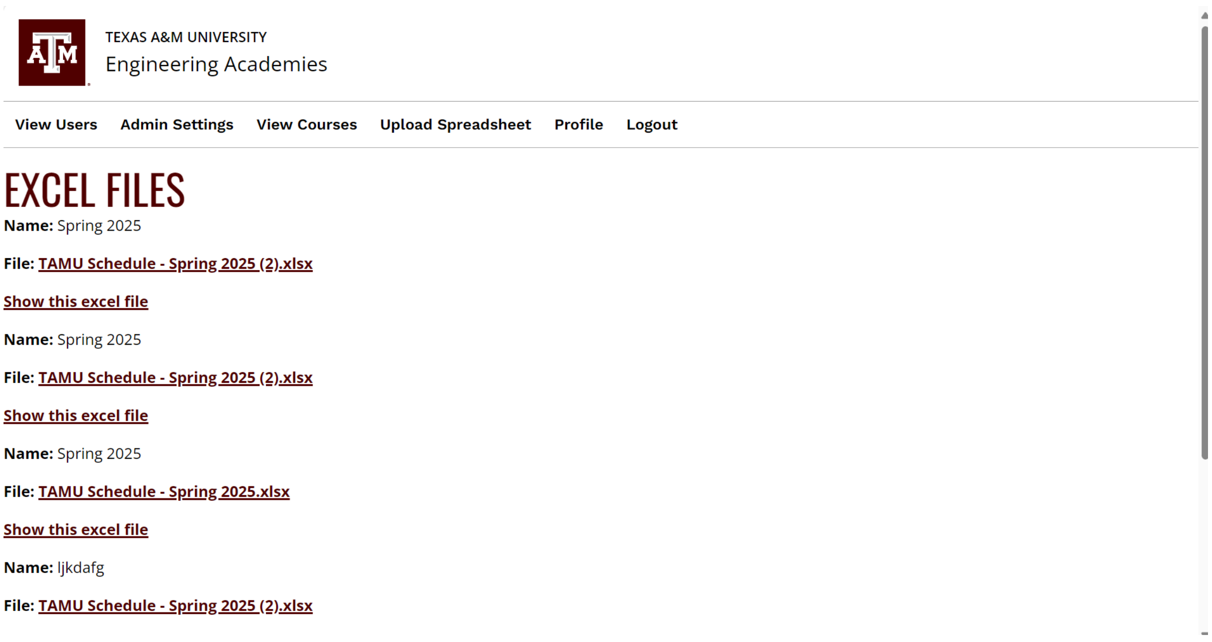

TEXAS A&M UNIVERSITY
Engineering Academies

View Users  Admin Settings  View Courses  Upload Spreadsheet  Profile  Logout

## Courses for term: 225S000

| Term | Course | Days | Start | End | Actions |
|---|---|---|---|---|---|
| 225S000 | CHEM 1109-001 | M | 2000-01-01 10:30:00 UTC | 2000-01-01 13:20:00 UTC | ✎ 🗑 |
| 225S000 | CHEM-1109-003 | W | 2000-01-01 10:30:00 UTC | 2000-01-01 13:20:00 UTC | ✎ 🗑 |
| 225S000 | CHEM 1309-001 | MW | 2000-01-01 09:00:00 UTC | 2000-01-01 10:20:00 UTC | ✎ 🗑 |
| 225S000 | MATH 2413-002 | MW | 2000-01-01 09:00:00 UTC | 2000-01-01 10:45:00 UTC | ✎ 🗑 |
| 225S000 | MATH 2414-009 | TTH | 2000-01-01 08:35:00 UTC | 2000-01-01 10:20:00 UTC | ✎ 🗑 |
| 225S000 | MATH 2414 016 | TTH | 2000-01-01 17:40:00 UTC | 2000-01-01 19:25:00 UTC | ✎ 🗑 |
| 225S000 | MATH 2415-006 | MW | 2000-01-01 12:00:00 UTC | 2000-01-01 13:45:00 UTC | ✎ 🗑 |
| 225S000 | MATH 2415-007 | MW | 2000-01-01 15:30:00 UTC | 2000-01-01 17:15:00 UTC | ✎ 🗑 |
| 225S000 | MATH 2420-004 | TTH | 2000-01-01 09:00:00 UTC | 2000-01-01 10:45:00 UTC | ✎ 🗑 |
| 225S000 | PHYS 2425-004 | MW | 2000-01-01 15:00:00 UTC | 2000-01-01 16:20:00 UTC | ✎ 🗑 |
| 225S000 | PHYS 2425-004 (Lab) | MW | 2000-01-01 16:30:00 UTC | 2000-01-01 17:50:00 UTC | ✎ 🗑 |
| 225S000 | PHYS 2425-005 | MW | 2000-01-01 13:30:00 UTC | 2000-01-01 14:50:00 UTC | ✎ 🗑 |
| 225S000 | PHYS 2425-005 (Lab) | MW | 2000-01-01 15:00:00 UTC | 2000-01-01 16:20:00 UTC | ✎ 🗑 |
| 225S000 | PHYS 2425-008 | TTH | 2000-01-01 13:30:00 UTC | 2000-01-01 14:50:00 UTC | ✎ 🗑 |
| 225S000 | PHYS 2425-008 (Lab) | TTH | 2000-01-01 15:00:00 UTC | 2000-01-01 16:20:00 UTC | ✎ 🗑 |
| 225S000 | PHYS 2426-004 | TTH | 2000-01-01 13:30:00 UTC | 2000-01-01 14:50:00 UTC | ✎ 🗑 |

127.0.0.1:3000


TEXAS A&M UNIVERSITY
Engineering Academies

View Users  Admin Settings  View Courses  Upload Spreadsheet  Profile  Logout

## EXCEL FILES

**Name:** Spring 2025

**File:** **TAMU Schedule - Spring 2025 (2).xlsx**

**Show this excel file**

**Name:** Spring 2025

**File:** **TAMU Schedule - Spring 2025 (2).xlsx**

**Show this excel file**

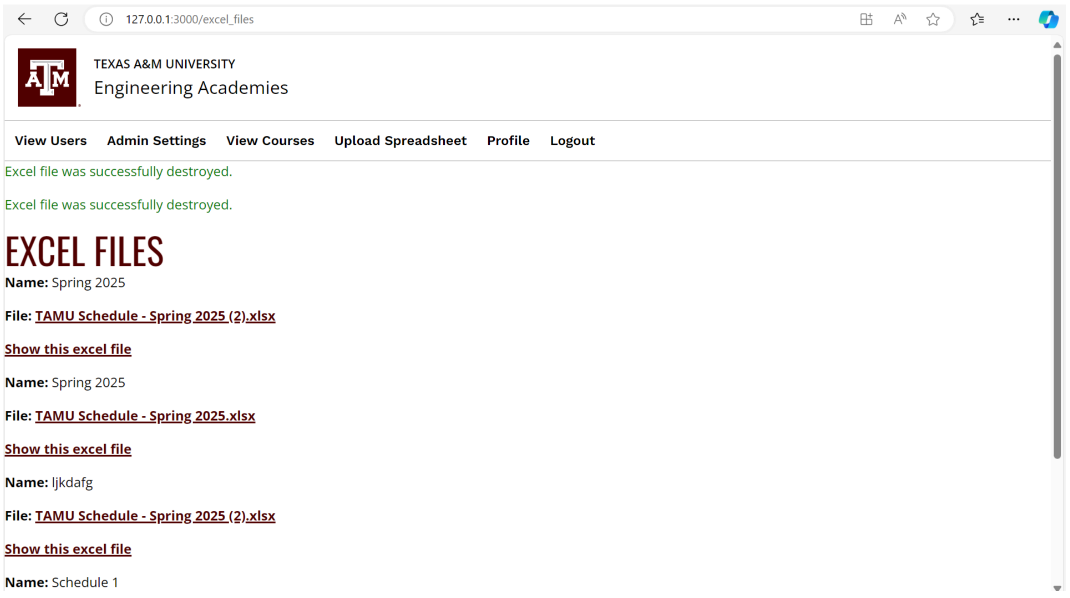
**Name:** Spring 2025

**File:** **TAMU Schedule - Spring 2025.xlsx**

**Show this excel file**

**Name:** ljkdafg

**File:** **TAMU Schedule - Spring 2025 (2).xlsx**


127.0.0.1:3000/excel_files

TEXAS A&M UNIVERSITY
Engineering Academies

View Users Admin Settings View Courses Upload Spreadsheet Profile Logout

Excel file was successfully destroyed.

Excel file was successfully destroyed.

EXCEL FILES

**Name:** Spring 2025

**File:** TAMU Schedule - Spring 2025 (2).xlsx

Show this excel file

**Name:** Spring 2025

**File:** TAMU Schedule - Spring 2025.xlsx

Show this excel file

**Name:** ljkdafg

**File:** TAMU Schedule - Spring 2025 (2).xlsx

Show this excel file

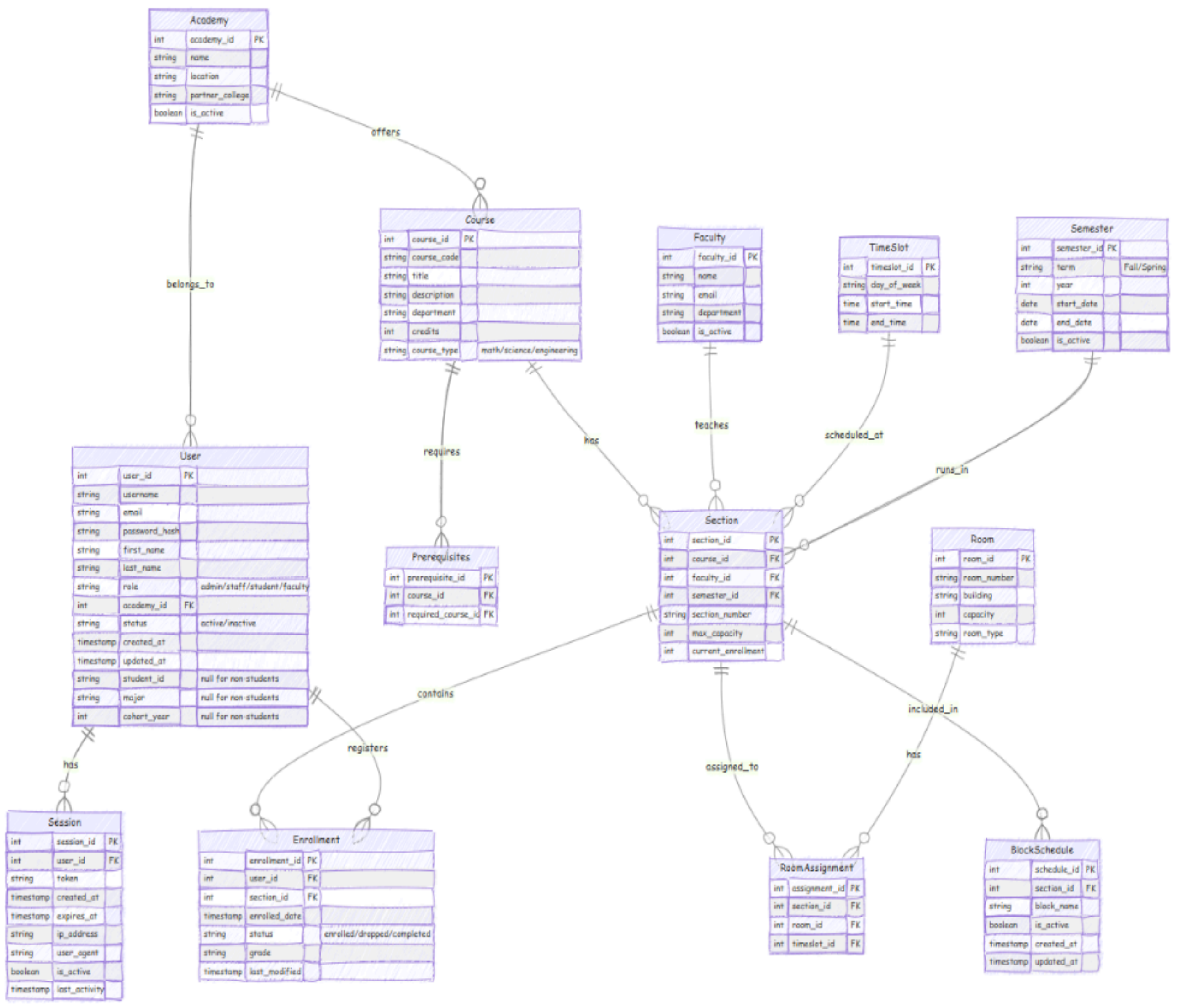
**Name:** Schedule 1

## 18. Sprint 3 documentation – 3 points

- As a developer,
  When I go to /documentation/Fall2024/ under the Github repository,
  I can see the documentation for sprint 3.
- Sprint plan documentation
- Sprint MVP documentation
- Sprint Retrospective documentation
- Sprint 3
- Status: Completed

## 19. Create Database schema – 5 points

- As a user I should be able to view the database schema and relation among the tables in an easy diagram form with well documentation.
- Sprint 4
- Status: Completed



**20. Add SSO Capability - 13 points**

- As a user
- I want to be able to use Outlook or Gmail
- So, I can easily sign into my TAMU account
- Sprint 2,3
- Status: Completed

**21. Add corequisite and type column to course table for schedule generation - 3 points**

- As a developer
- I need to add two columns to the course table
- So that I can use the schedule generation algorithm
- Also needed to move functions related to the course controller from the seed file into the controller (improperly placed) and integrate into the create function to properly create courses.
- Sprint 4
- Status: Completed

**22. Generate Schedule Page - 5 points**

- As a developer implemented a user interface for course selection and schedule generation. This feature allows users to view selected courses and initiate the schedule creation process with a single button click.
- Sprint 1
- Status: Completed

**23. Initialize Ruby on Rails and Cucumber and Rspec - 3 points**

- As a developer set up the foundational Ruby on Rails environment for the application. This task also included configuring Cucumber for behaviour-driven development, rspec for TDD ensuring a robust testing framework from the project's inception.
- Sprint 1
- Status: Completed

**24. SSO and OAuth Integration - 13 points**

- As a developer implemented Single Sign-On (SSO) functionality, enabling users to log in via Gmail. Used omniauth ruby gem and set up credentials for allowing users to sign in.
- Sprint 2,3
- Status: Completed

**25. Deploy to Heroku - 8 points**

- As a developer deployed the application to Heroku, making it accessible to users in a live environment. This deployment involved configuring necessary settings like database and ensuring smooth operation in the cloud-based platform.
- Sprint 1
- Status: Completed

**26. Integrate Code Climate - 5 points**

- Added Code Climate to the repository to maintain high code quality standards. This integration helps the team adhere to best practices and continuously monitor the health of the codebase.
- Sprint 2
- Status: Completed

**27. Create Session and User Tables - 3 points**

- As a user I should be able to see the database schema for managing user sessions and account information including roles. This ensures secure and efficient handling of user data and permissions throughout the application.
- Sprint 3,4
- Status: Completed

**28. Develop Student and Admin Views - 8 points**

- As a user should be able to see distinct interfaces tailored for both student and administrator roles. This differentiation in views enhances usability by providing role-specific functionalities and information access.
- Sprint 3,4
- Status: Completed

**29. Integrate with database – 5 points**

- Sprint 1
- Create tables for courses and users
- Status: Completed

**30. Spreadsheet upload feature – 10 points**

- Sprint 1
- Create spreadsheet parser
- Create view to see recently uploaded courses
- Status: Completed

**31. Add roles and user-roles – 10 points**

- Sprint 3
- Create roles for access rights to different areas of the application
- Integrate roles and users through user-roles
- Status: Completed

**32. UI Layouts – 5 points**

- Sprint 4
- Created navbar
- Created footer
- Status: Completed

**33. Updated application UI to match TAMU recommendations – 5 points**

- Sprint 4
- Matched UI of our website based on TAMU website's UI
- Status: Completed

**34. Create pages controller to act as switchboard – 2 points**

- Sprint 4
- Added a controller to allow for pages to be linked more smoothly
- Status: Completed

**35. General bug fixes – Handle invalid user login – 6 points**

- Sprint 4
- Fixed bugs for dashboard
- Fixed a glitch that was not allowing users to log in
- Status: Completed

**36. Add admin requirement for user views except for profile view – 10 points**

- Sprint 4
- Add admin constraint checker
- Add profile view and route
- Add admin ability to edit/destroy a user
- Status: Completed

**37. Add application settings for admin**

- Sprint 4
- Add features to the admin settings view
- Status: Completed

## 3. If Legacy project describe your understanding - NA

## 4. List who held each team role, e.g. Scrum Master, Product Owner. Describe any changes in roles during the project

**Sprint1- Gifted Macaw**
Product Owner: Ryann Lu; Scrum Master: Chengyan Tsai
**Sprint2-Ecstatic Woodchuck**
Product Owner: Chengyan Tsai; Scrum Master: Aaron Jones
**Sprint3-Hungry Hamster**
Product Owner: William-David Vanderpuye; Scrum Master: Mahima Bhatt
**Sprint4-Flamboyant Manatee**
Product Owner: Junhyuk Lee; Scrum Master: Adithi Srinath

## 5. Summaries of each sprint. The vast difference of completed points between some of the sprints is due to completion of previous sprint's stories (carryovers and spillovers from last sprint)

Gifted Macaw Sprint 1
Accomplished:

- Generate Schedule Page
- Add a landing page
- Initialize the Ruby on Rails repo and configure Cucumber
- Deploy the app to Heroku
- DB schema discussion
- UI mockup discussions

Total Points: 34
Ecstatic Woodchuck Sprint 2
Accomplished:

- Make each classes object
- Upload the excel spreadsheet
- Integrate Code climate to repo
- UI Mock-up
- System Structure Diagram

Total Points: 54
Hungry Hamster Sprint 3
Accomplished:

- Show basic schedule timetable
- Store the spreadsheet data on the database
- Add tokens for logging in
- Add SSO Capability
- Connect login page to student dashboard
- Set typical classes grouping on the web
- Set block generating conditions on the web
- User pick rest of classes
- Display Class Blocks
- Display Class Details
- Create session and user tables
- Student and Admin views
- Sprint documentation

Total Points: 91
Flamboyant Manatee Sprint 4
Accomplished:

- Retrieve data from database
- Application makes possible blocks with conditions
- Generate possible blocks
- Add prerequisite and type column to course table for schedule generation
- Create form for completed courses
- Fix UI for header and footer
- Add UI for landing page
- Fix UI for dashboard
- Bug Fix - Dashboard

Total Points: 58

## 6. User Story Points Table:

| Team Member | Total Points |
|---|---|
| Ryann Lu | 34 |
| Chengyan Tsai | 31 |
| Aaron Jones | 53 |
| William-David Vanderpuye | 18 |
| Mahima Bhatt | 41 |
| Junhyuk Lee | 49 |
| Adithi Srinath | 45 |

## 7. List of customer meeting dates, and description of what happened at the meetings, e.g. what software/stories did you demo.

<u>10/10/24</u>

Reviewing progress with Prof. Shana Shaw (client)
UI mockup
Basic page (“Selected courses”)
Login page for landing

Need to include number of students per block
Hierarchy for given classes?

Engineering, math, science class - necessary every term
216 - need Pre Cal and Cal 1 for this class - and need ENGR 102
102 finished physics and chemistry

Are all classes in the ACC system worried about location? No, all classes in one location, roughly

First come first serve? Yes, for scheduling, when section fills, they must pick a different block schedule - open registration
Update options when section is full

<u>10/24/24</u>

Make sure any excel format can be parsed?

1311 or 1309 can be used as a foundational math course

Only need Chem 2 for BMEN, CHEM, MSEN majors

No student info, on the system

Can choose chem or physics

12 hours minimum but no maximum every term

Math, engineering, and science required every term

Potentially include having 1401 or high school physics as a prereq for physics 1

Probably don't need to assign them physics and chemistry at same time

IM is instruction method - 1 means lecture, 2 means lab, and 12 is a combined (back-to-back) lecture-lab block

Need syn (Synonym) number from ACC (which is similar to a CRN at A&M)

11/07/24
Add algorithm to have classes a student has already taken
Make sure multiple admins have access to admin features
Nice-to-have: Live course capacity

11/22/24
Full demo day with client
Demonstrated login feature, user profile feature, and schedule generation feature
Described final steps of integration

## 8. Explain your BDD/TDD process, and any benefits/problems from it.

BDD and TDD were used to ensure high-quality, bug-free code and meeting user requirements. With BDD, we wrote clear scenarios with Cucumber, specifying the expected behavior in plain language derived from the user stories. With TDD, we implemented unit tests before writing code, ensuring that each feature was developed with a clear goal. We did our best to follow what we learned in class.

This approach brought several benefits. By requiring full test coverage before merging code into the main branch, we avoided bugs and conflicting code, ensuring a clean and stable codebase. The tests also acted as a safety net, allowing us to refactor and improve our code, especially during integration, knowing that the tests would catch any regressions.

We did have problems from practicing this new process. Writing tests before implementing features slowed the initial development pace, especially for complex functionality as we were growing used to the new methods. Additionally, maintaining tests during requirement changes or refactoring added more work to the table. Despite these hurdles, the time invested in testing was justified by the significant reduction in bugs and smoother integration, leading to a more reliable and maintainable codebase. We only ever had major issues with the codebase when untested code was merged accidentally.

## 9. Discuss your configuration management approach. Did you need to do any spikes? How many branches and releases did you have?

Our configuration management approach for implementing changes was:

1. Create stories during Sprint plans
2. Verify direction of stories with client
3. Decide on points during Sprint plans
4. Implement feature on a local branch
5. Receive confirmation and approval with verification testing
6. Merge change to main branch

We did not have to do any spikes. We had over 60 branches from each feature being verified before being merged into the main branch.

**10. Discuss any issues you had in the production release process to Heroku.**
While deploying, we faced a db setup issue. The database had issues setting up but finally fixed and the url was showing up in heroku config. Also chose a non-essentials plan but switched to basic after realizing it wasn't the most economical tier.

**11. Describe the tools/Gems you used, such as GitHub, CodeClimate, SimpleCov, and their benefits and problems.**

**GitHub** - We used GitHub for version control and overall file management for our project.

Benefits: Good platform that allows for a streamlined way for a software team to work on a project together. Version control allows for easy reversion of edits if necessary. Branches allow for different group members to work on individual mini projects before committing changes to the main branch.

Problems: We had some issues with our main branch - in particular certain parts of the project got deleted after certain commits. We also had some issues with the branches getting behind in terms of commits, meaning our branches were desynced.

**CodeClimate** - We used CodeClimate to ensure code quality, maintainability, complexity, and ensure our code was DRY.

Benefits: Gives an easy way to ensure that committed code does not have code smells. This forced us to make sure our code was written well.

Problems: Maintainability reports were not always clear on exactly how to increase maintainability

**SimpleCov** - Used for coverage analysis - Used to test how much code tests cover.

Benefits: Allows for viewing coverage

Problems: Only reports the latest test, does not store percentage coverage for previous tests

**Taiga** - We used Taiga to create and present user stories with points and descriptions for each user story.

Benefits: Created an organized way for storing user stories

Problems: User interface can be hard to deal with, not always intuitive where certain user stories are. User story points must be added to one part of the total (out of UX, Design, Front, or Back) in order for points to be added (as opposed to being able to add points to the total).

## 13. Project Repository and Deployment Process

This Ruby on Rails application is designed to use Single Sign-On (SSO) through the Omniauth gem with Google credentials for authentication. The application is deployed on Heroku. Below, we will outline the repository contents and the deployment process, including the necessary scripts and configurations to ensure smooth deployment and operations.

**Repository Structure**

The repository contains the following key directories and files to support deployment and SSO integration:

1. app/: Contains the main application code, including models, views, controllers, and helpers.
    - controllers/: Includes authentication logic through Omniauth in sessions_controller.rb.
    - helpers/: Contains an authentication_helper.rb to manage the SSO authentication flow.
    - views/: Contains the views rendered by the application, including login and callback views for Google SSO.
2. config/: Holds configuration files for Rails and Heroku deployment.
    - config/initializers/omniauth.rb: The Omniauth initializer, where Google credentials (client ID, client secret) are configured to enable the SSO functionality.
    - config/secrets.yml: Stores sensitive data, including the Omniauth client ID and client secret. This file is not tracked in the repository for security reasons. These secrets are provided through Heroku environment variables. Or you can set .env variables. We have created the secret in the Github repo.
3. Gemfile: Specifies the gems used by the application, including:
    - omniauth and omniauth-google-oauth2 for handling authentication with Google.
    - pg for PostgreSQL support (required for Heroku).
    - rails, puma, sass-rails, and other necessary gems for a Rails app.
4. Gemfile.lock: Records the exact versions of gems to ensure consistency between development, staging, and production environments.
5. Rakefile: Includes tasks to run database migrations, seed data, etc., during deployment.
6. app/assets/: Contains static assets, including logos or other images used in the authentication process.
7. README.md: Provides instructions for setting up the project locally and deploying it to Heroku. It includes:
    - Prerequisites for local development (Ruby, Rails, PostgreSQL).
    - Setup instructions for Omniauth with Google credentials.
    - Deployment steps using Heroku.
    - How to set admin/student view and permissions.
8. config.ru: Used for Rack-based deployment, ensuring the app is correctly loaded by Heroku.
9. .gitignore: Excludes files such as log/, tmp/, and node_modules/ from being tracked by Git.

**Deployment Process**

To deploy the application to Heroku, follow these steps:

1. Clone the repository (if not already cloned):

```
$ git clone https://github.com/tamu-edu-students/EA-Block-Scheduling.git
$ cd your_repo_name
```

2. Set up the environment for local development:

Ensure you have Ruby and Rails installed.

Install dependencies:

```
$ bundle install
```

Set up the database (using PostgreSQL in production):

```
$ rake db:create
$ rake db:migrate
```

3. Set up Google OAuth credentials:
    - Obtain your Google Client ID and Google Client Secret from the Google Developer Console.

Add these credentials as Heroku environment variables:

```
$ heroku config:set GOOGLE_CLIENT_ID=your-client-id
$ heroku config:set GOOGLE_CLIENT_SECRET=your-client-secret
```

4. Deploy to Heroku:

Create a Heroku app (if not already created):

```
$ heroku create
```

Push your code to Heroku:

```
$ git push heroku main
```

Database Migration on Heroku: After deployment, run the migrations on Heroku:

```
$ heroku run rake db:migrate
```

5. Access the app: Once deployed, access the app via the Heroku URL provided, e.g., https://your-app-name.herokuapp.com.

**14. Links to your Project Management tool page, public GitHub repo, and Heroku deployment, as appropriate. Make sure these are up-to-date.**

Github Repo: https://github.com/tamu-edu-students/EA-Block-Scheduling
Taiga (project page): https://tree.taiga.io/project/aaronjones05-block-scheduler/wiki/home
Deployed App: https://ea-block-scheduler-4fecd886e389.herokuapp.com/

**15. Links to your presentation video and demo video.**

Demo Video: https://youtu.be/mM4b1CGJPZ0