

Team 1: Software Engineering Project

Final Presentation & Demo

Science Alliance Portal

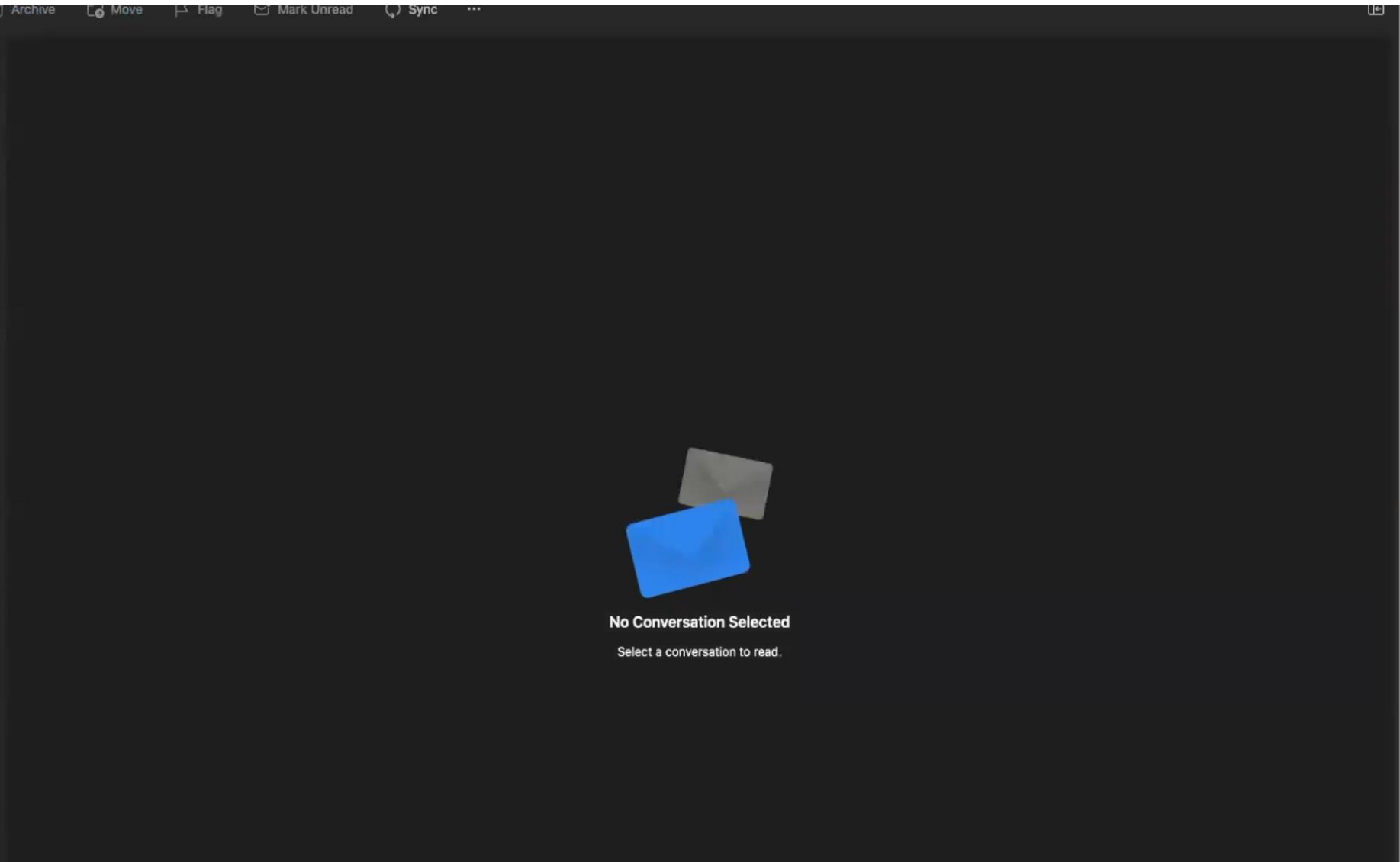
Ahmed Z., Mahmood A., Le N., Murugesan S., Shaeffer E.

May 10, 2021

Background

- ❖ Who we made it for
- ❖ Why we made it
- ❖ What does it do





Product Recap

Administrative tool to help

- **Help provide research opportunities to undergraduates**
- **Recruit future Science Alliance Interns**
- **Track Participation within Science Alliance**

Technology Used

- We used Python django as the backend framework for our application
- The application is deployed on Microsoft Azure
- To have a working email functionality for features like Signup verification we used SendGrid
- We used pylint to make sure that our code has perfect code quality following PEP8 standards
- We used vanilla Javascript and frameworks like Momentjs for improved UX



Process Strengths & Weaknesses

Strengths

- Pair Programming
- Unit Tests
- Quality Review/ Pull Requests

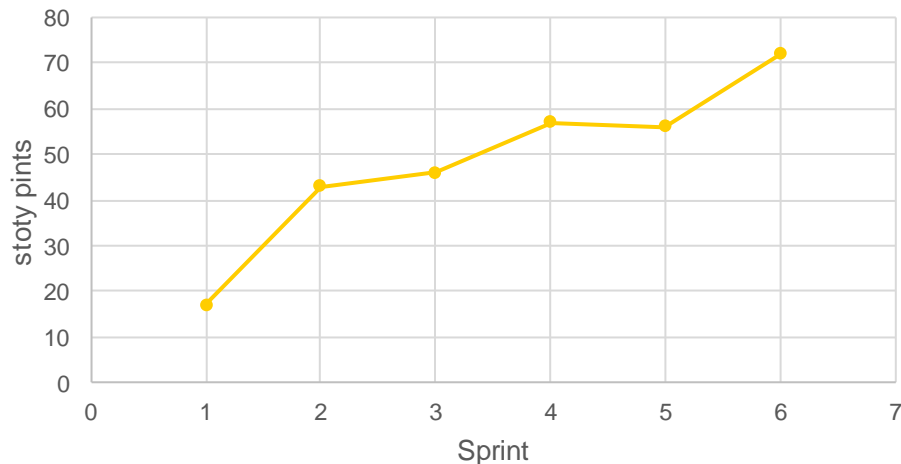
Weaknesses

- Feature testing
- Performance testing
- Not fully testing possible inputs

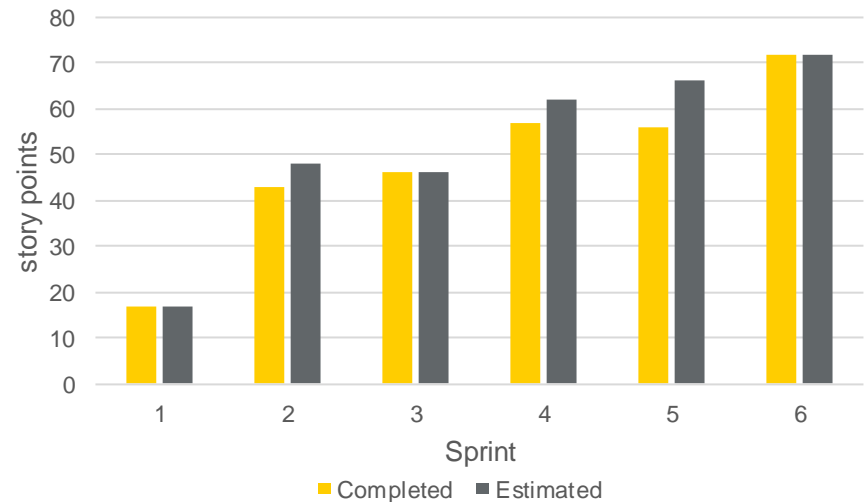
Sprint Velocity and Productivity

- Achieved the MVP at Sprint 4
- All high-priority user stories successfully implemented
- User stories are categorized into Epics: user authentication, dashboard, data collection, email, events, insights, support, and documentation
- Team's capacity increases after each sprint

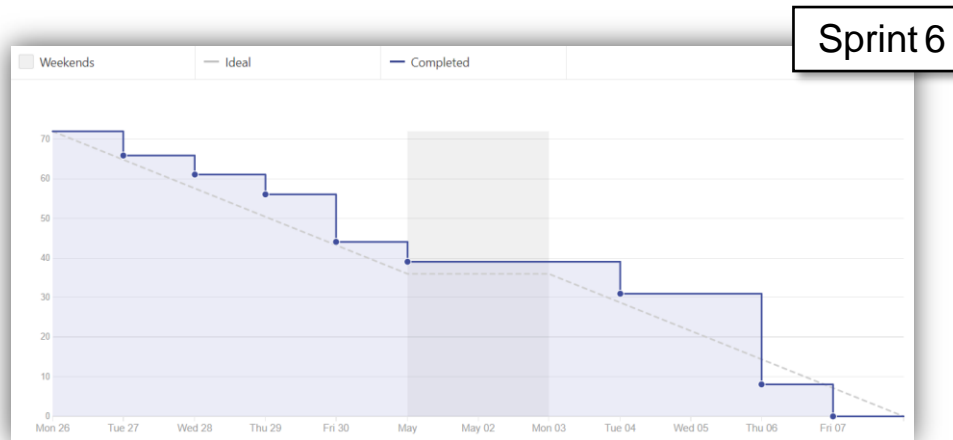
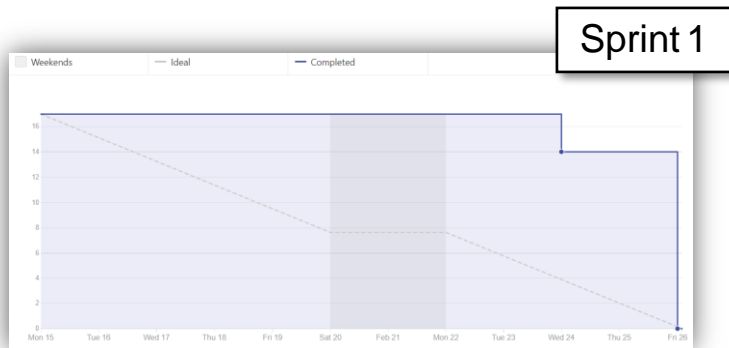
Team's capacity in 6 sprints



Velocity tracking



Burndown Report & Code Coverage



Name	Stmts	Miss	Cover
sap/__init__.py	0	0	100%
sap/forms.py	13	0	100%
sap/templatetags/__init__.py	0	0	100%
sap/templatetags/filter.py	18	3	83%
sap/tokens.py	10	0	100%
sap/upload_resource_to_azure.py	25	0	100%
sap/urls.py	10	0	100%
sap/views.py	667	36	95%
sap/views_v2.py	488	16	97%
sap/views_v3.py	532	38	93%
TOTAL	1763	93	95%
Ran 132 tests in 68.809s			
OK			

Code coverage

Problems Encountered

- Merge Conflicts
- Coverage report
- Code Coverage – line exclusions
- GitHub artifacts – 1Gb for org – all 3 teams – reached limit –
cronscript with GitHub action to delete old artifacts – GitHub
actions – 7days old artifacts deleted
- Azure site – slow website--web server running US-west
- Database in US-central – to diff locations client->us-west->us-
central->us-west... fixed by US-central

Lessons Learned

- Not delay deployment
- Not delay tasks till the end of sprint
- Merge to production branch more often
- Sprint story points estimation
- Don't delete migration files

Team 1: Software Engineering Project

Live Demo, Q&A

Thank you!

