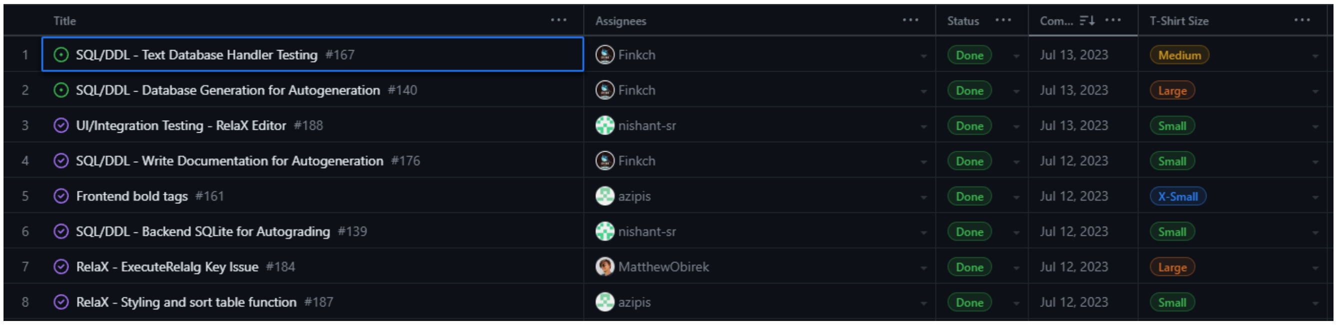
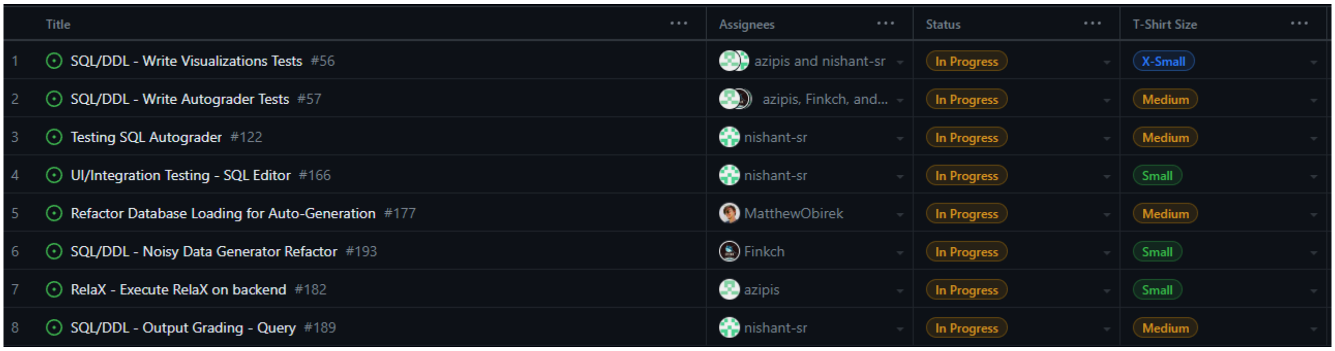
Project 3: Automating Database Question Generation and Marking - Team A
July 12 - July 14
Task Summary

Completed since last meeting:



	Title	Assignees	Status	Com...	T-Shirt Size
1	SQL/DDL - Text Database Handler Testing #167	Finkch	Done	Jul 13, 2023	Medium
2	SQL/DDL - Database Generation for Autogeneration #140	Finkch	Done	Jul 13, 2023	Large
3	UI/Integration Testing - RelaX Editor #188	nishant-sr	Done	Jul 13, 2023	Small
4	SQL/DDL - Write Documentation for Autogeneration #176	Finkch	Done	Jul 12, 2023	Small
5	Frontend bold tags #161	azipis	Done	Jul 12, 2023	X-Small
6	SQL/DDL - Backend SQLite for Autograding #139	nishant-sr	Done	Jul 12, 2023	Small
7	RelaX - ExecuteRelalg Key Issue #184	MatthewObirek	Done	Jul 12, 2023	Large
8	RelaX - Styling and sort table function #187	azipis	Done	Jul 12, 2023	Small

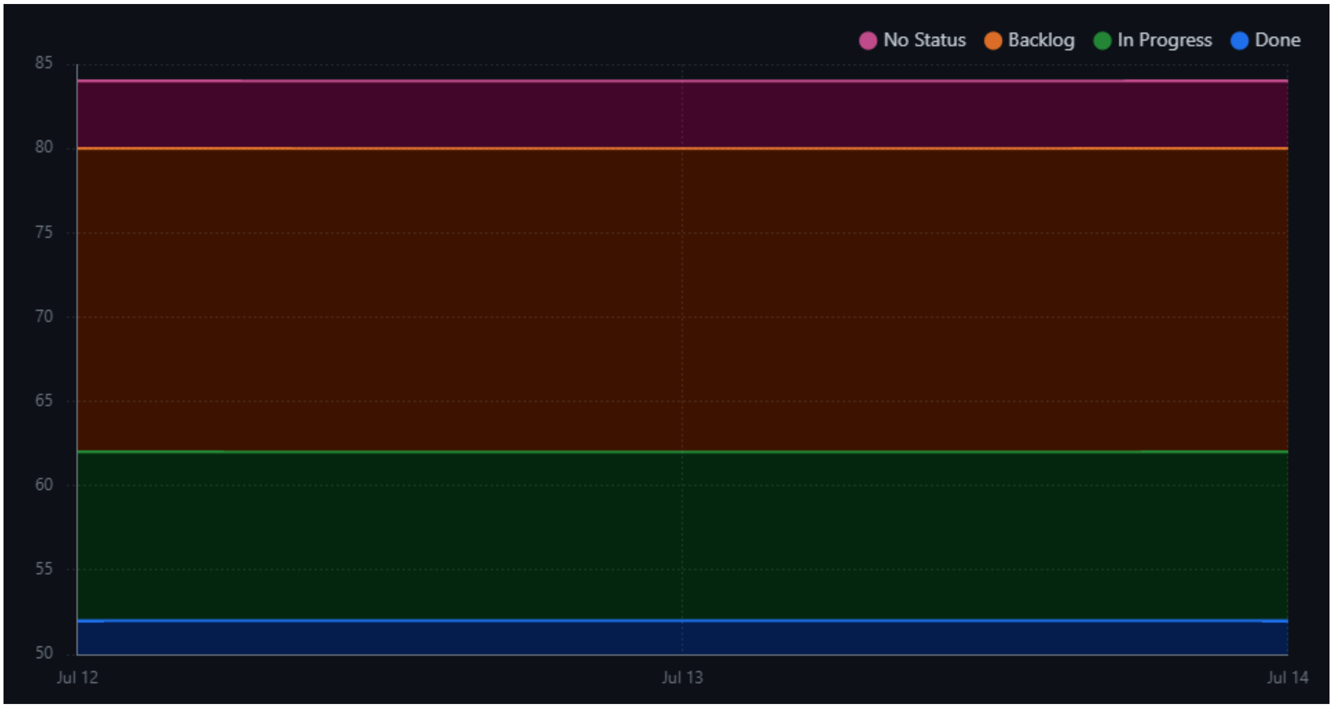
In progress:



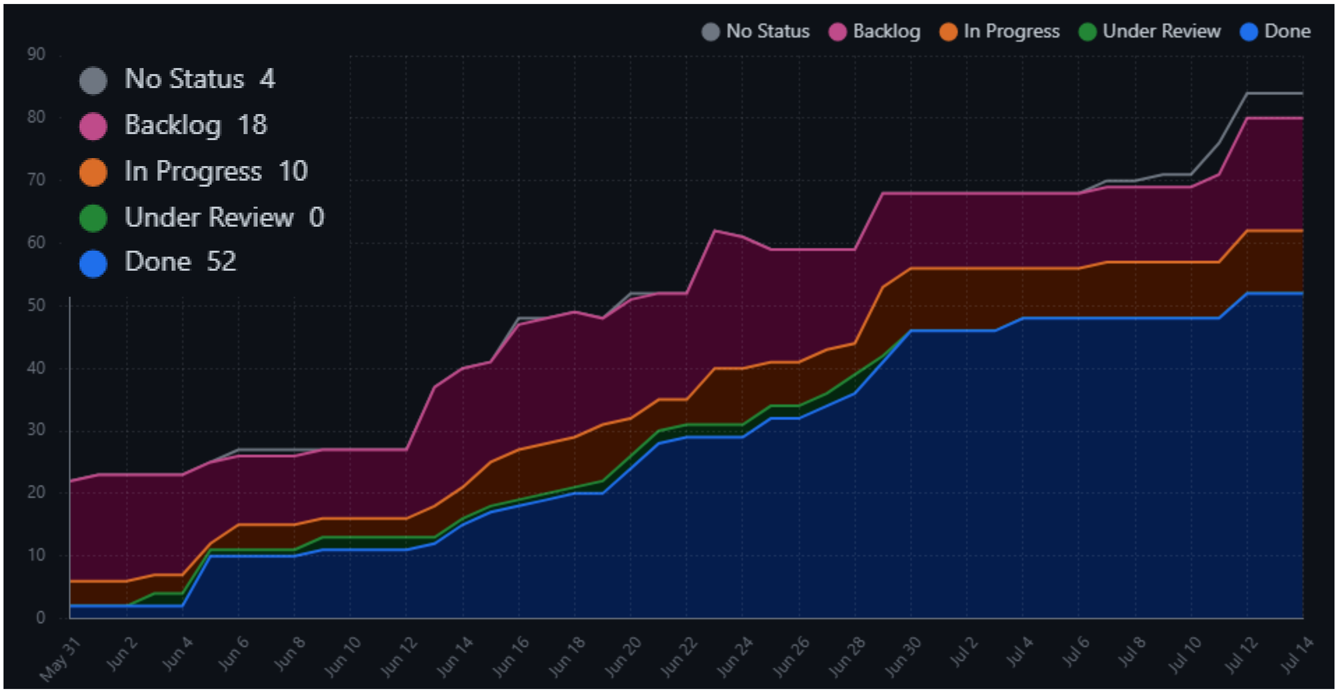
	Title	Assignees	Status	T-Shirt Size
1	SQL/DDL - Write Visualizations Tests #56	azipis and nishant-sr	In Progress	X-Small
2	SQL/DDL - Write Autograder Tests #57	azipis, Finkch, and...	In Progress	Medium
3	Testing SQL Autograder #122	nishant-sr	In Progress	Medium
4	UI/Integration Testing - SQL Editor #166	nishant-sr	In Progress	Small
5	Refactor Database Loading for Auto-Generation #177	MatthewObirek	In Progress	Medium
6	SQL/DDL - Noisy Data Generator Refactor #193	Finkch	In Progress	Small
7	RelaX - Execute RelaX on backend #182	azipis	In Progress	Small
8	SQL/DDL - Output Grading - Query #189	nishant-sr	In Progress	Medium

Burnup since last meeting:

Note: Github insights not working accurately. Burnup shows that now changes occurred since July 12 but you can see above that's incorrect.



Burnup to date:



REQUIREMENTS		Type of Test	Pass or Fail	Contributor

Functional				N=Nishant,
				A=Andrei,
				S=Skyler,
				M=Matthew

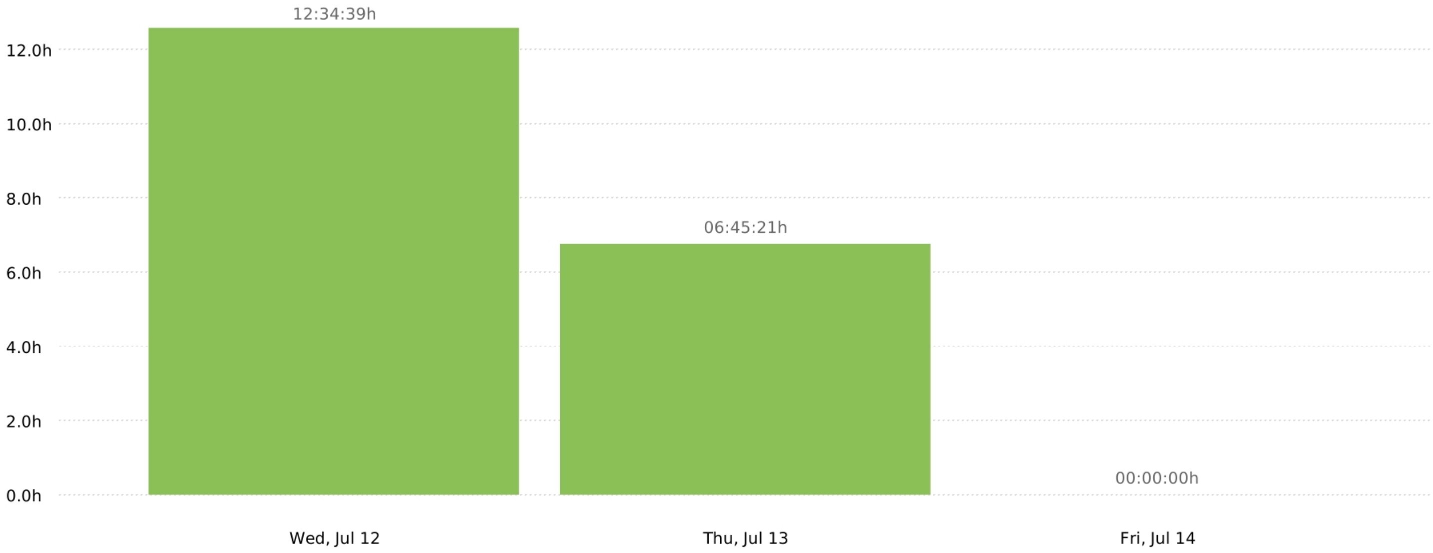
1.1	System will allow for relational algebra statements to be entered.	UI Testing Integration Testing	Fail Fail	
1.2	System will show visualizations of the resulting entered statement prior to submission.	UI Testing Integration Testing	Fail Fail	
1.3	System will automatically mark the relational algebra questions once submitted.	Unit Testing	Fail	
1.4	System will allow for DDL/SQL code to be entered. System will show resulting tables of queries prior to submission.	UI Testing Integration Testing	Pass Pass	N, N
1.5	System will automatically mark the DDL/SQL questions once submitted.	Unit Testing	Fail	
1.6	Student will be able to see the correct answer if the professor has allowed for the correct answer to be displayed after the question is submitted.	Unit Testing Integration Testing UI Testing	Fail Fail Fail	
1.7	Professor will be able to set whether the correct answer will be displayed after the question is submitted.	Unit Testing UI Testing	Fail Fail	
1.8	Professor will be able to see the correct answer.	UI Testing	Fail	
Non-Functional				
2.1	The system will support all COSC 304 users simultaneously – about 200 students.	Performance Testing	Fail	
2.2	The system will ensure data integrity and preservation so that no data is lost upon submission.	Performance Testing	Fail	
2.3	The system will display entered queries within 3 seconds at scale and under optimal conditions.	Performance Testing	Fail	
2.4	The system will return automarked submissions within 5 seconds at scale and under optimal conditions.	Performance Testing	Fail	
2.5	The user interface will match existing software used for COSC 304.	UI Testing	Fail	
Technical Requirements				
3.1	Rebuild RelaX editor and calculator into PrairieLearn	UI Testing Integration Testing	Fail Fail	
3.2	Frontend: JavaScript, HTML, CSS	UI Testing	Pass	N
3.3	Backend: Python, Node.JS	Unit Testing	Pass	N
3.4	Write JavaScript code that takes in SQL/DDL statements and displays appropriate table results	Integration Testing	Pass	N
3.5	Write Python code that automatically marks submitted data and returns the students grade	Unit Testing	Fail	

Summary report

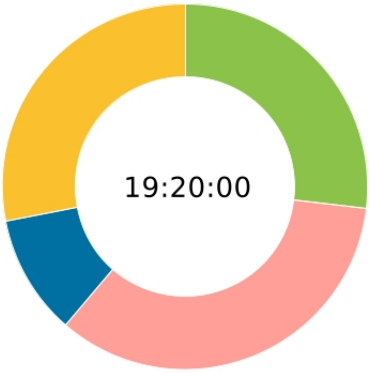
Clockify

07/12/2023 - 07/14/2023

Total: 19:20:00

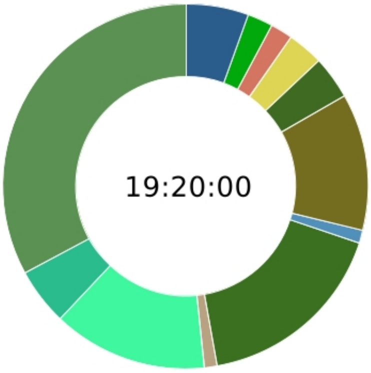


User



●	Andrei Zipis	05:25:44	28.08%
●	Matthew Obirek	02:02:53	10.59%
●	Nishant Srinivasan	06:39:35	34.45%
●	Skyler A.	05:11:48	26.88%

Description



●	Team Meeting	06:22:06	32.94%
●	#161 Frontend bold tags	00:57:49	4.98%
●	ui testing	02:39:32	13.75%
●	project board management	00:12:47	1.10%
●	Random Table Generation	03:19:30	17.20%

●	Dashboard	00:12:29	1.08%
●	#187 RelaX - Styling and sort table function	02:21:36	12.21%
●	Report Meeting	00:43:57	3.79%
●	fix broken ui tests	00:36:54	3.18%
●	#179 Product table generation PR	00:22:47	1.96%
●	Logs + Dashboard	00:25:33	2.20%
●	SQLiteDB Grading	01:05:00	5.60%

User / Description	Duration
Andrei Zipis	05:25:44
#161 Frontend bold tags	00:57:49
Dashboard	00:12:29
#187 RelaX - Styling and sort table function	02:21:36
#179 Product table generation PR	00:22:47
Team Meeting	01:31:03
Matthew Obirek	02:02:53
Team Meeting	02:02:53
Nishant Srinivasan	06:39:35
ui testing	02:39:32
Team Meeting	00:55:52
project board management	00:12:47
Report Meeting	00:43:57
fix broken ui tests	00:36:54
Logs + Dashboard	00:25:33
SQLiteDB Grading	01:05:00
Skyler A.	05:11:48

Random Table Generation	03:19:30
Team Meeting	01:52:18