Project 3: Automating Database Question Generation and Marking - Team A
July 19 - July 21
Task Summary

Completed since last meeting:

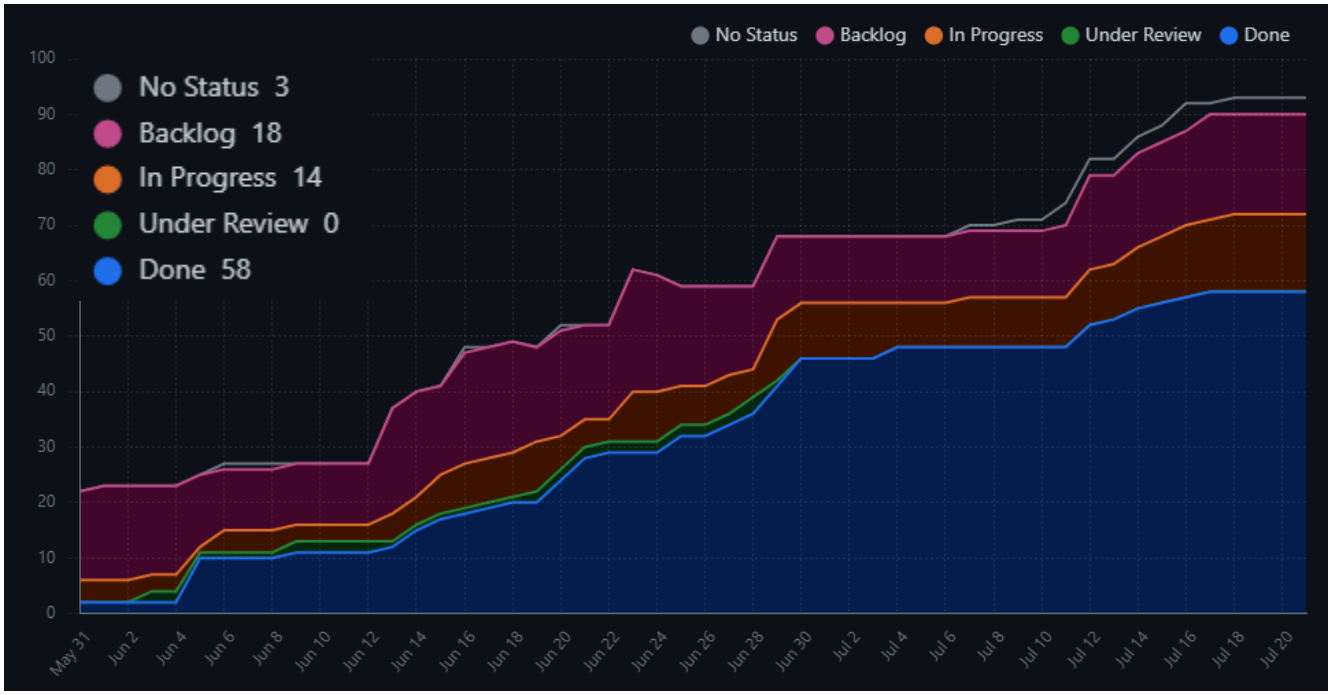
	Title	Assignees	Status	Com...	T-Shirt Size
1	Testing SQL Autograder #122	nishant-sr	Done	Jul 20, 2023	Small
2	Database and Table Generalization #210	Finkch	Done	Jul 20, 2023	Medium

In progress:

	Title	Assignees	Status	T-Shirt Size
1	RelaX - Autogenerator - Database Data #29	Finkch and Matthe...	In Progress	
2	SQL/DDL - Write Visualizations Tests #56	azipis and nishant-sr	In Progress	X-Small
3	SQL/DDL - Write Autograder Tests #57	azipis, Finkch, and...	In Progress	Medium
4	SQL/DDL - UPDATE-style Generation Parameterization #156	Finkch	In Progress	Medium
5	SQL/DDL - DELETE-style Generation Parameterization #157	Finkch	In Progress	Medium
6	SQL/DDL - QUERY-style Generation Parameterization #158	Finkch	In Progress	Large
7	RelaX - Execute RelaX on backend #182	azipis	In Progress	Large
8	SQL/DDL - Output Grading - Query #189	nishant-sr	In Progress	Medium
9	SQL/DDL - Output Grading - Create #190	nishant-sr	In Progress	Small
10	SQL/DDL - Output Grading - Insert #206	nishant-sr	In Progress	Small
11	SQL/DDL - Output Grading - Update #204	nishant-sr	In Progress	Small
12	SQL/DDL - Output Grading - Delete #205	nishant-sr	In Progress	Small
13	Show expected output #212	nishant-sr	In Progress	Small

Burnup since last meeting:

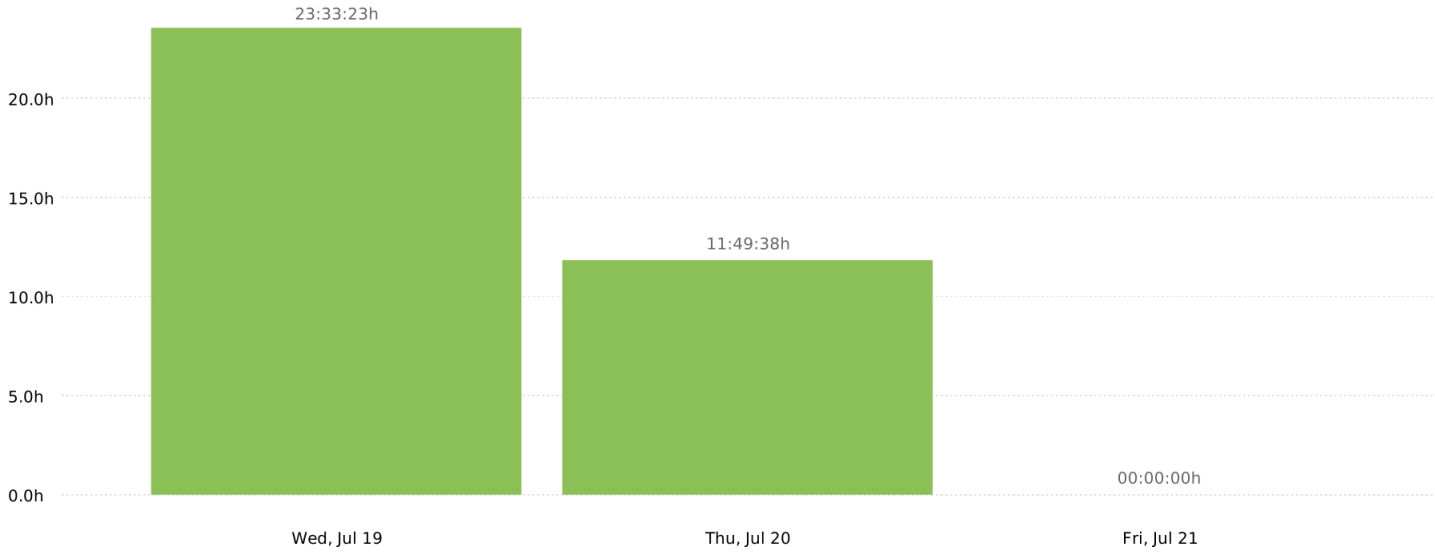
Burnup to date:



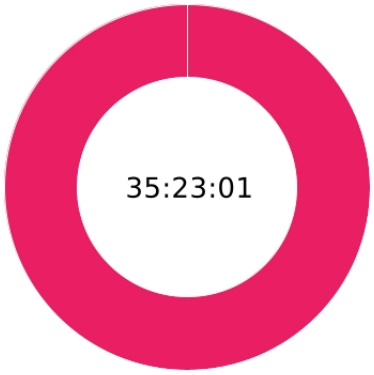
Summary report

07/19/2023 - 07/21/2023

Total: 35:23:01

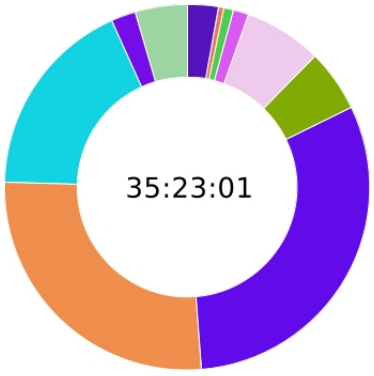


Project



Project 3: Automating Database Question Generation - Dr. Lawrence	35:23:01	100.00%

Description



SQLiteDB Grading Testing	01:37:55	4.61%
Report Meeting	00:45:03	2.12%
#182 RelaX - Execute RelaX on backend	06:18:54	17.85%
Generalizing Databases and Data Generation	09:27:06	26.71%
Team Meeting	10:55:48	30.89%

Code Cleans	01:56:11	5.47%
SQLiteDB Grading	02:26:39	6.91%
Expected Output Preview	00:29:15	1.38%
logs	00:16:31	0.78%
Question Autogen	00:11:11	0.53%
Database Autogeneration - table referencing	00:58:28	2.75%

Project / Description	Duration
Project 3: Automating Database Question Generation - Dr. Lawrence	35:23:01
SQLiteDB Grading Testing	01:37:55
Report Meeting	00:45:03
#182 RelaX - Execute RelaX on backend	06:18:54
Generalizing Databases and Data Generation	09:27:06
Team Meeting	10:55:48
Code Cleans	01:56:11
SQLiteDB Grading	02:26:39
Expected Output Preview	00:29:15
logs	00:16:31
Question Autogen	00:11:11
Database Autogeneration - table referencing	00:58:28

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	Pass	
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	