Team Undefined

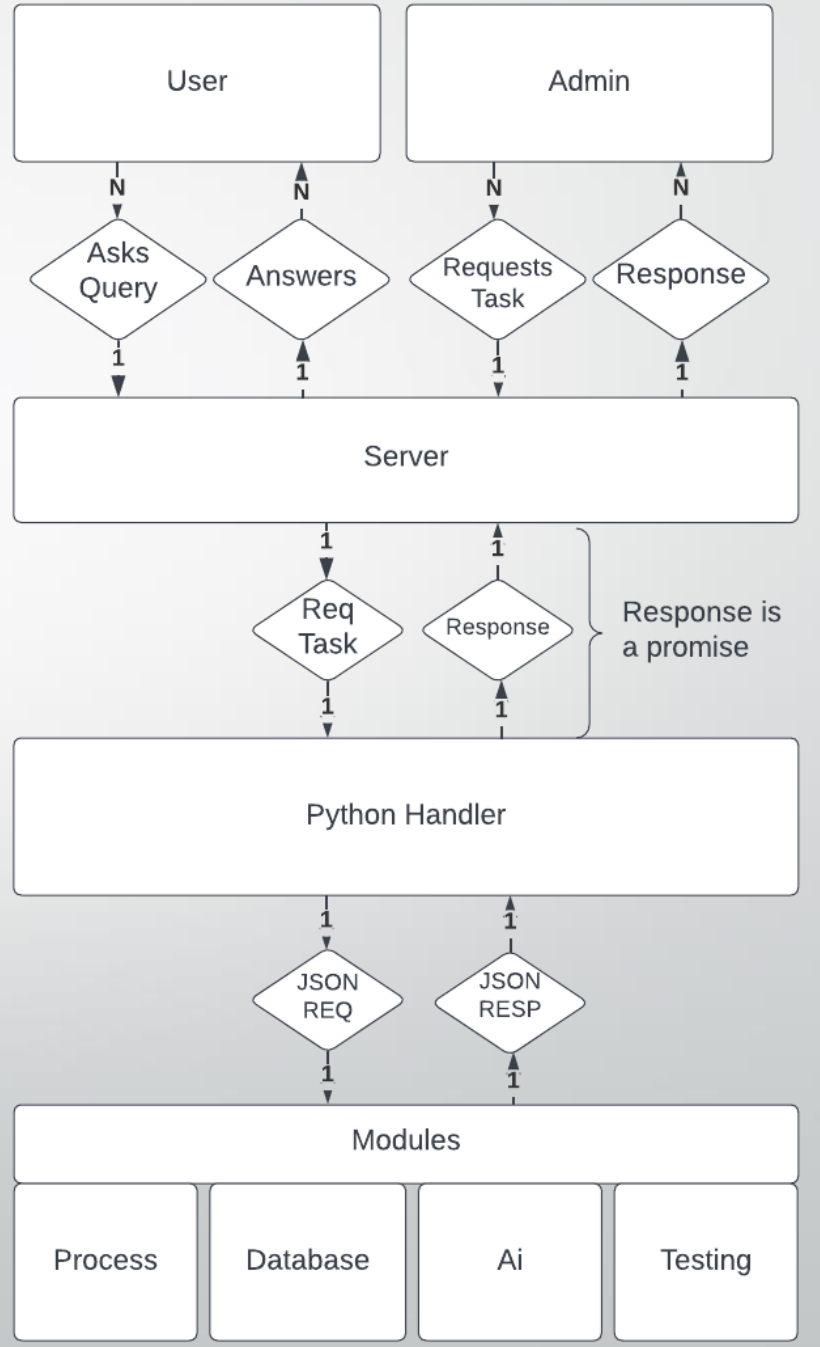
Main Features

- Responsive system: Each component is run on a unique thread, so the user/admin requests get an appropriate response.
- Admin page: To see component status, test results and Scraped data.
- Additional info on where the Ai's answer came from.

Links

GitHub	https://github.com/d3v3lopingCod3/Cosc4p02ChatbotProject
Product Backlog	https://miro.com/app/board/uXjVOVYcNxA=/?invite_link_id=823826101299
Progress Reports + Project Proposal	https://github.com/d3v3lopingCod3/Cosc4p02ChatbotProject/tree/main/Documents

System Entity Relationship Diagram



Developmental Team

Name	Role	Id	Brock Email	Github
Eddy Su	Scraping	6459705	es17zj@brocku.ca	eddysu123@gmail.com
Manvendrasinh Rana	Scraping	6137228	mr16gr@brocku.ca	manvendrarana@hotmail.cc
Aman Braich	Front-End	6511679	ab18ew@brocku.ca	braich_aman3@yahoo.ca
Raghav Bhardwaj	Front-End	6548580	rb18nr@brocku.ca	raghavmanc@gmail.com
Sawyer Fenwick	Back-End	6005011	sf15zx@brocku.ca	sawyerfenwick@gmail.com
Sager Kudrick	Back-End	5919170	sk15xm@brocku.ca	sagerkudrick@hotmail.com
Rikveet Singh Hayer	AI, Scraping	6590327	rh18vo@brocku.ca	rh18vo@gmail.com

Agile Software Development

Agile Development Methods

- Maintain Simplicity
  - Focus on simplicity in both the software being developed and in the development process
- People, Not Process
  - Skills of the Dev Team should be recognized and exploited
  - Team Members should be left to develop their own ways of working without prescriptive processes
- Incremental Development
  - The Software is developed in increments, with the customer specifying the requirements to be included in each increment

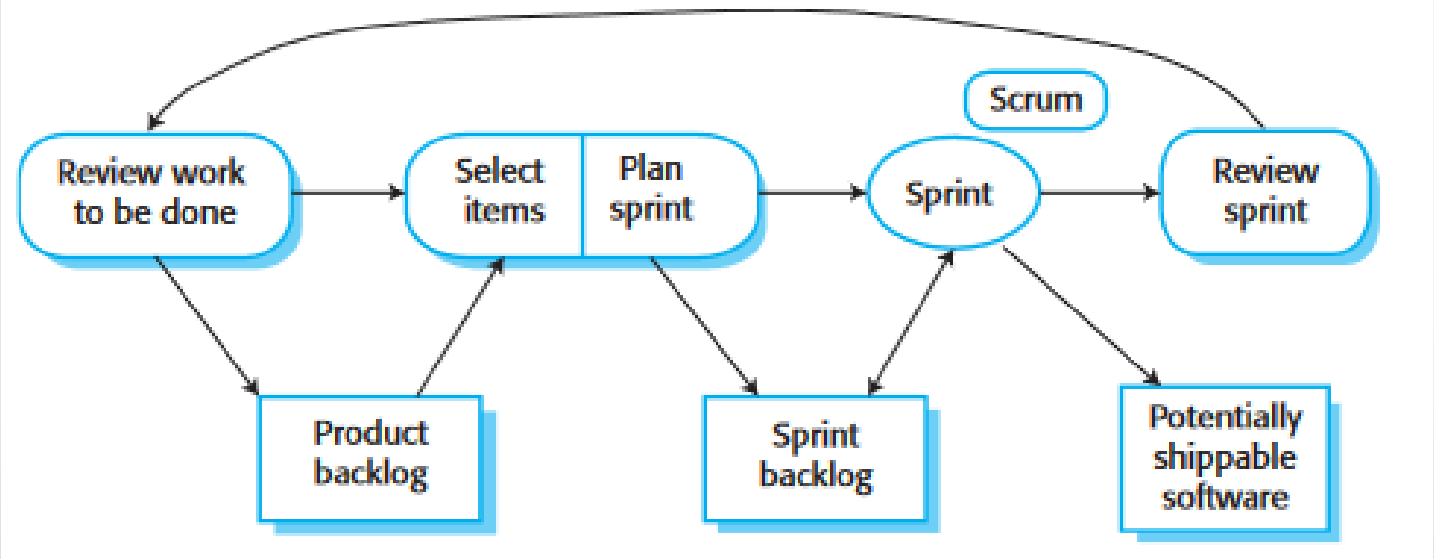
"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."
(http://agilemanifesto.org/principles.html, 2001)

Agile Development Techniques

- Simple Design
  - Enough design is carried out to meet the current requirements and no more
- Pair Programming
  - Developers work in pairs, checking each other's work and providing the support to always do a good job
- Collective Ownership
  - Pairs of developers work on all areas of the system so that no islands of expertise develop and all the developers take responsibility for all of the code. Anyone can change anything.
- Incremental Planning
  - Requirement are recorded on story cards and the stories to be included in a release are determined by the time available and their relative priority. The developers break these stories into development tasks.
- Refactoring
  - Reorganization of class hierarchy to remove duplicate code, tidying up and renaming of attributes and methods, and the replacement of similar code sections.

Sawyer Fenwick

Agile Project Management

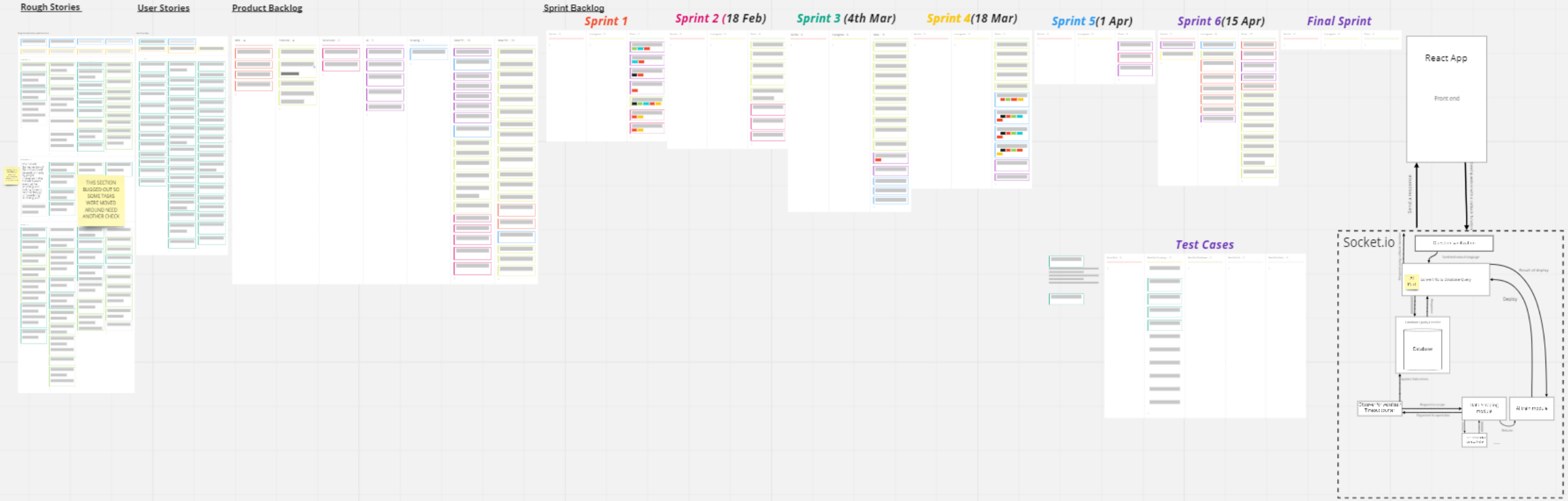


Scrum Sprint Cycle

Sawyer Fenwick

Backlogs

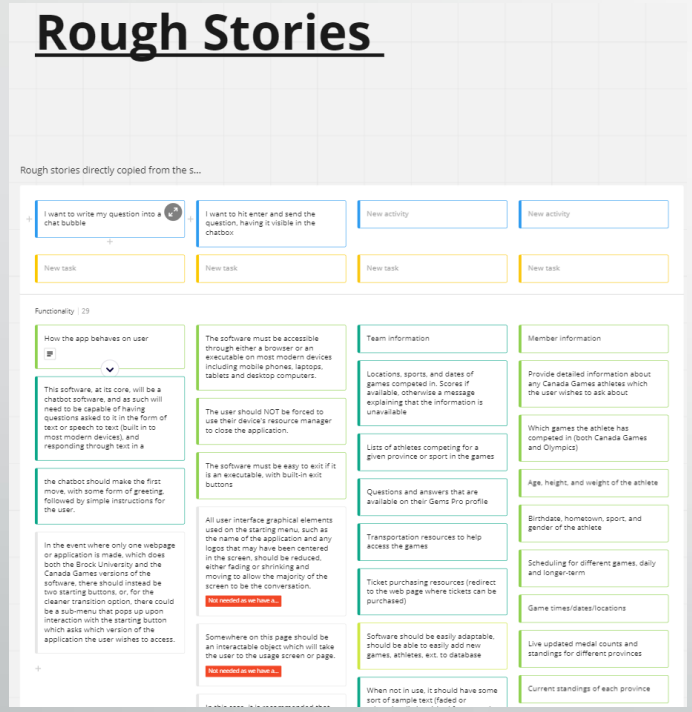
Miro



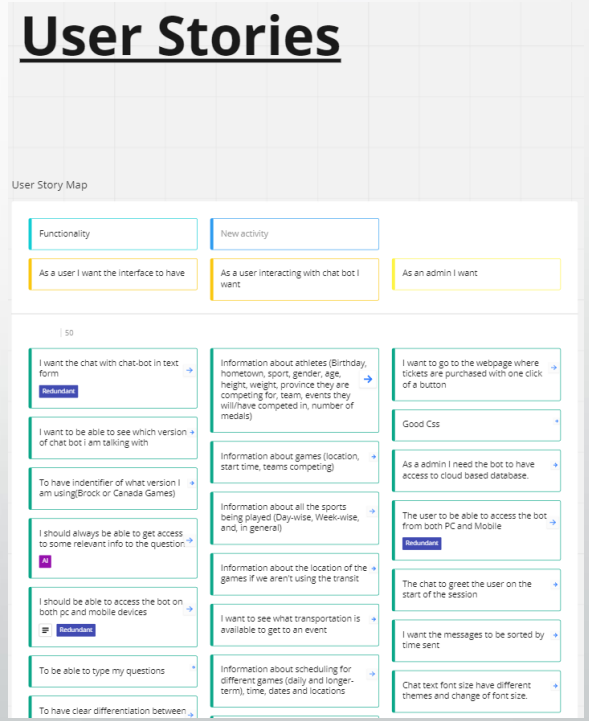
Manvendra Rana

Stories Management

Rough Stories

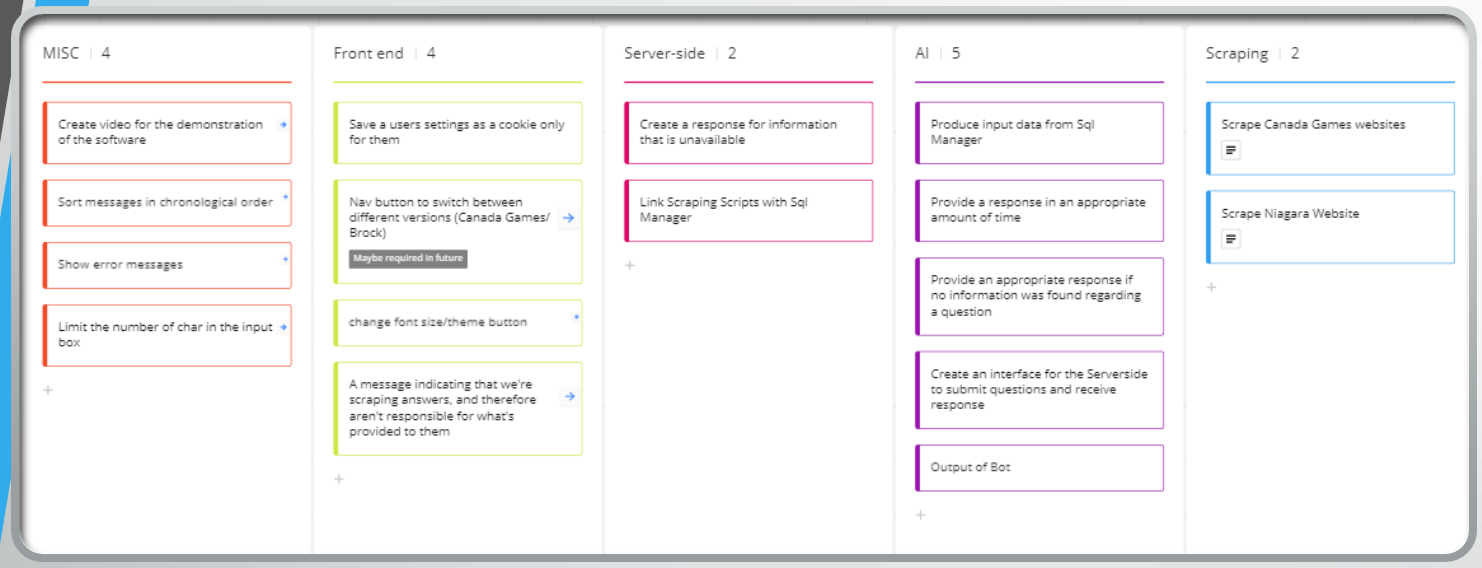


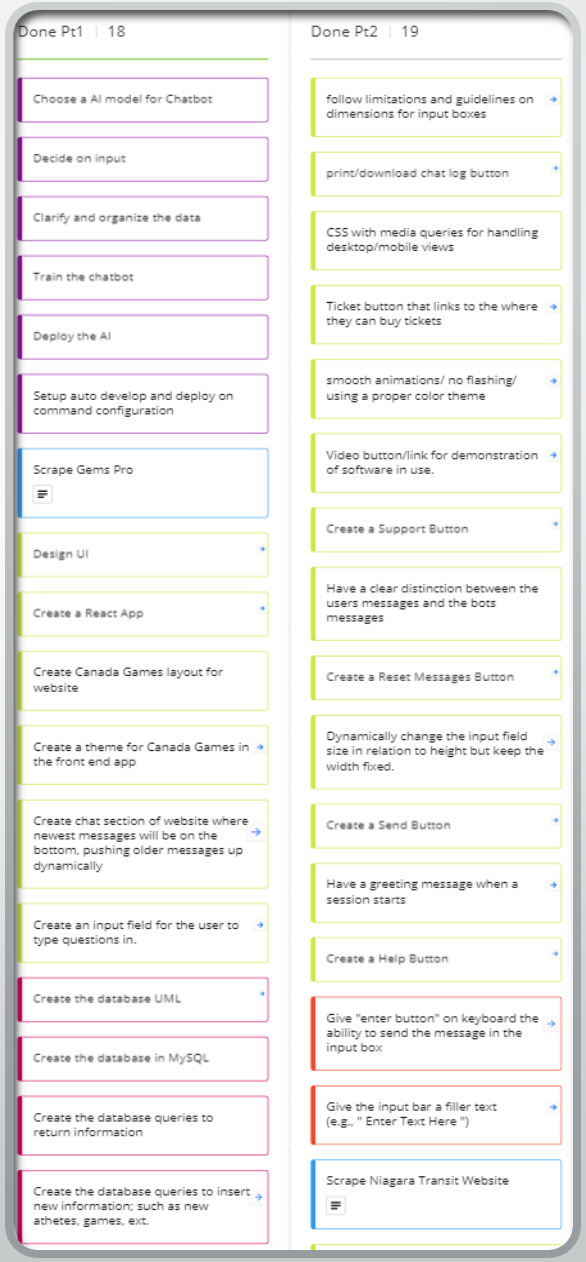
User Stories



Manvendra Rana

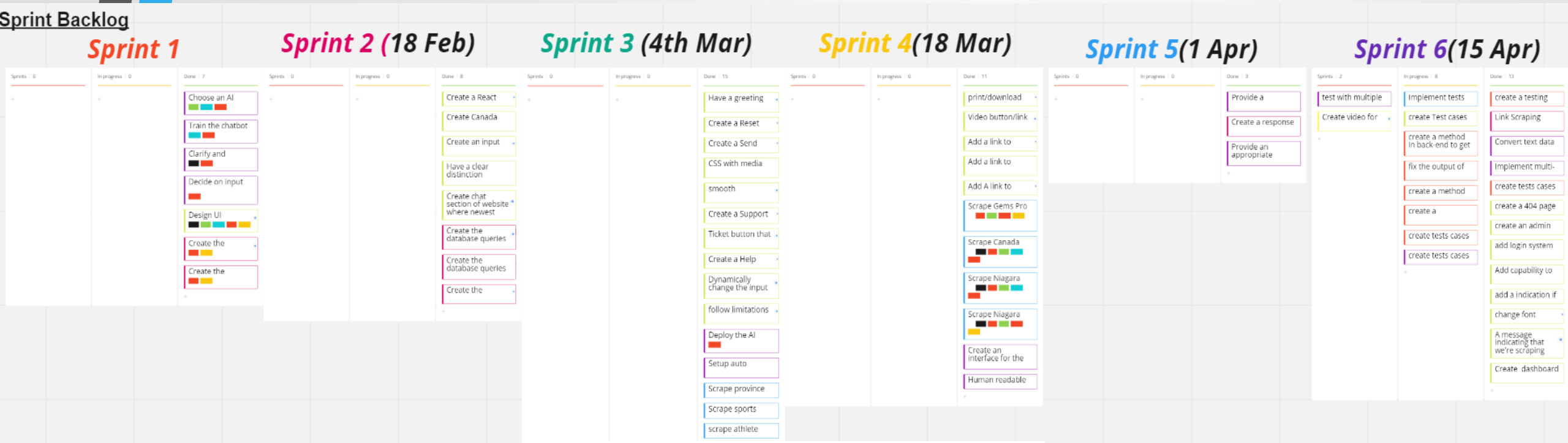
Product Backlog

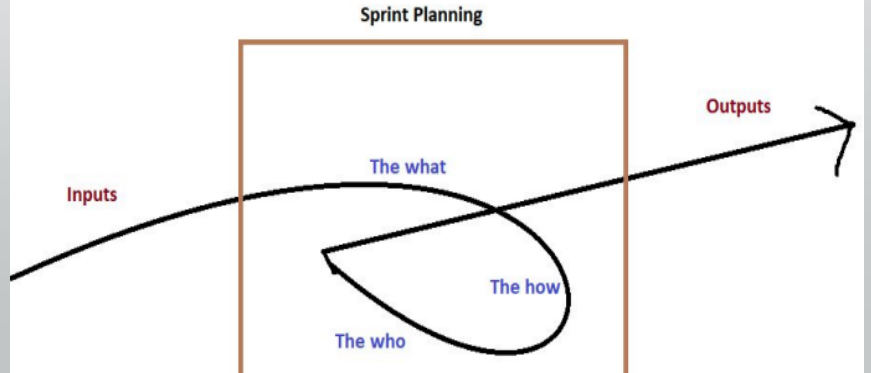




Manvendra Rana

Sprint Backlog





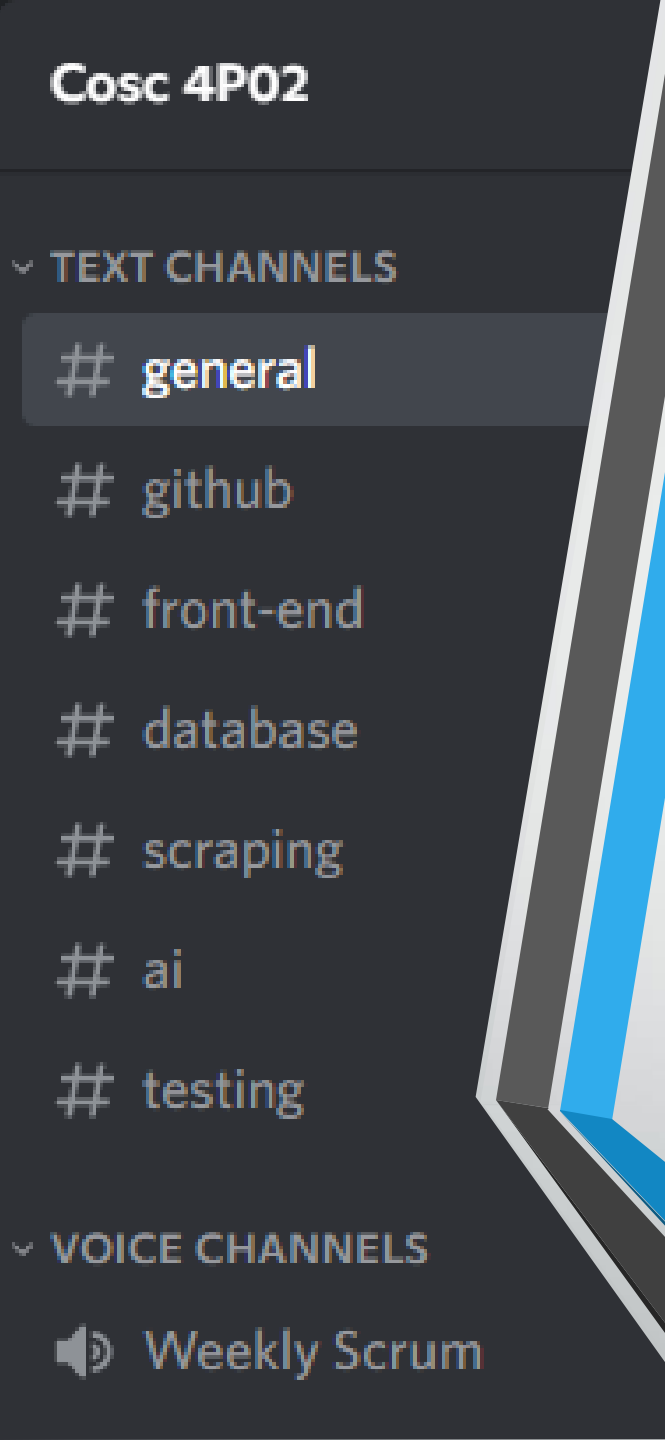
Manvendra Rana

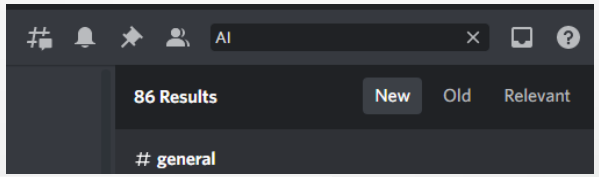
Communication Mediums

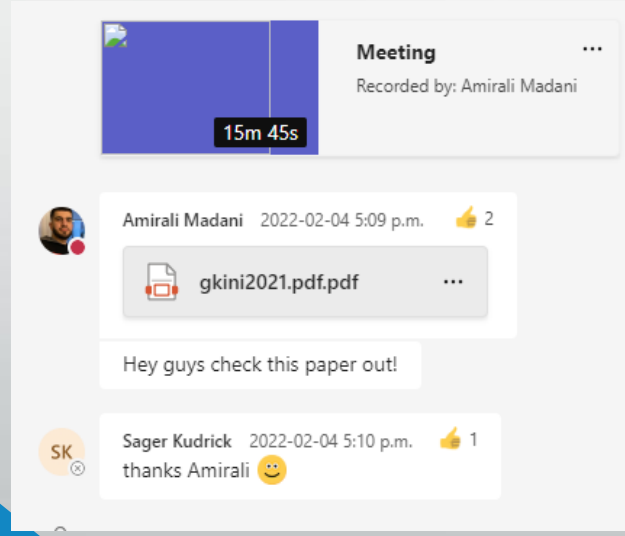
Communication Software

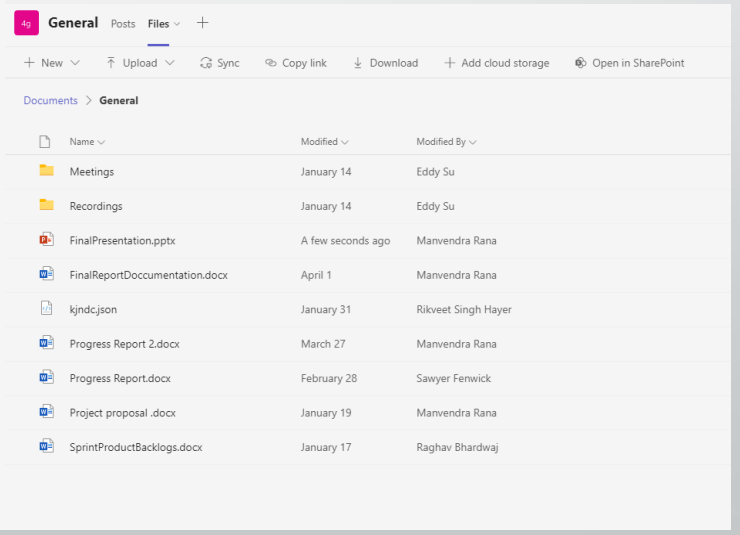
Discord

Microsoft Teams/OneDrive









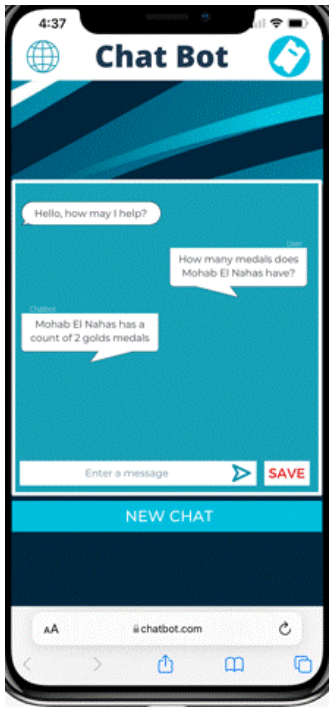
Meetings

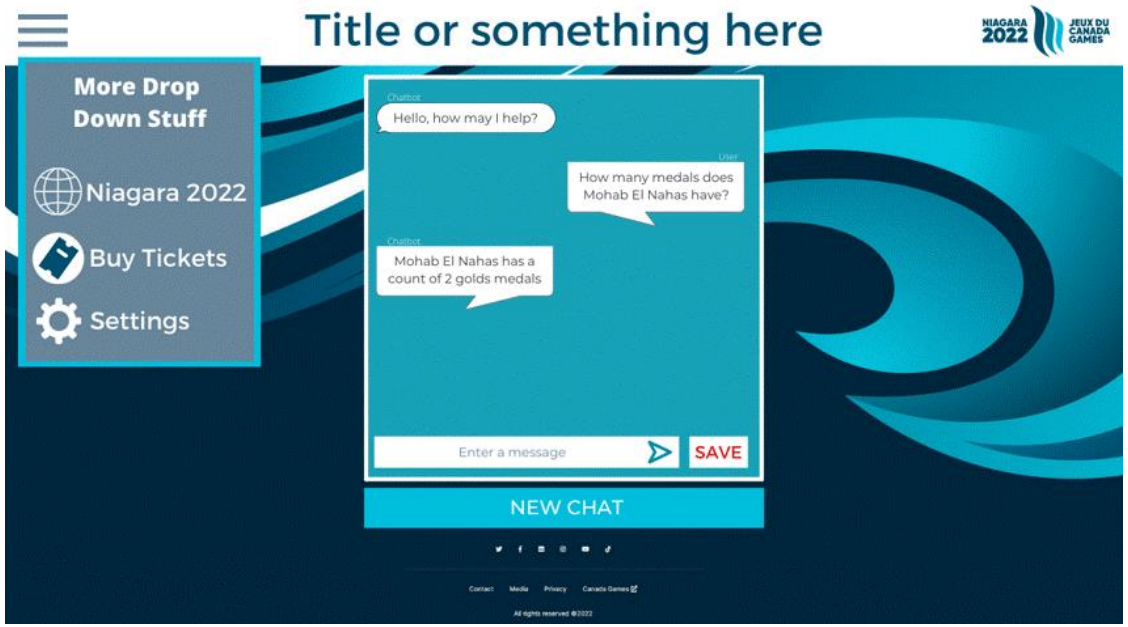
Meetings

- Meetings on Tuesday and Friday
- Main agenda:
  - Choosing tasks from backlog to work on during sprint
  - Occasional informal 1 on 1 meetings between scrum master, team members

Front-End

Mockup → Final





Sprint 1 → Sprint 6

React JS

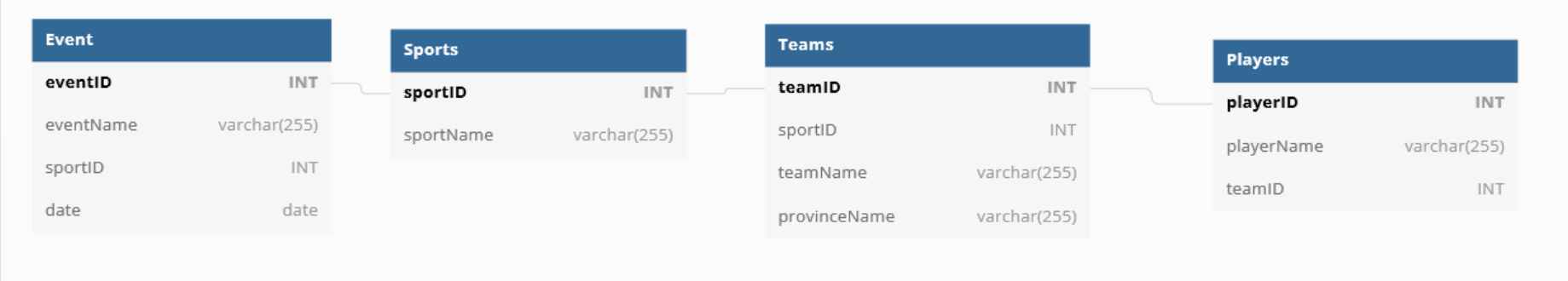
Socket Io

Aman

Back-End

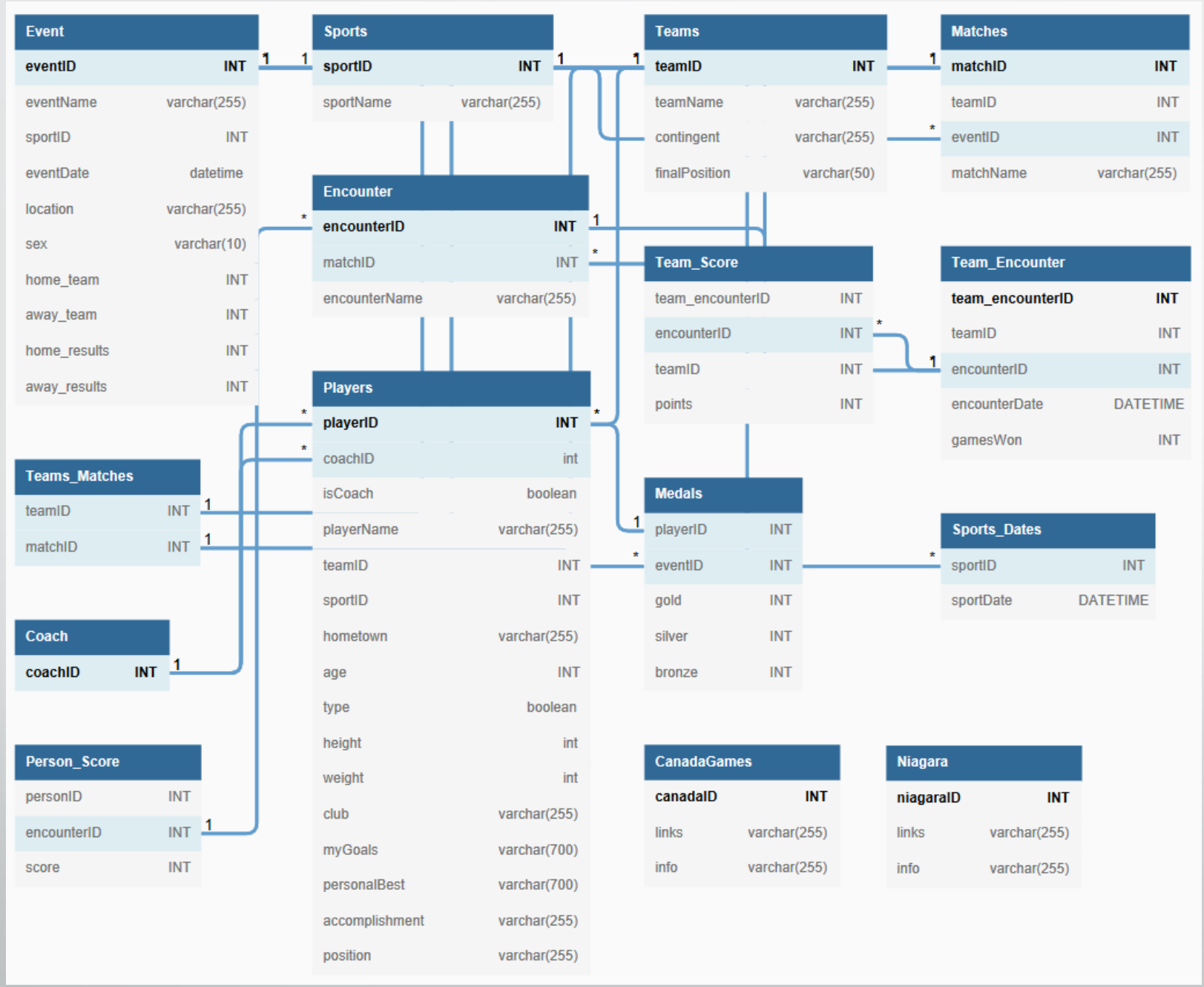
Database

SQL Database



Version 1 DB

Sawyer Fenwick



Version 2 DB

Sawyer Fenwick

Version 3 DB

Sawyer Fenwick

Teams	
id	INT
name	varchar(255)
gold	INT
silver	INT
bronze	INT
total	INT

Players	
id	INT
name	varchar(255)
age	INT
team	varchar(255)

Document Info	
key	varchar(255)
url	varchar(255)
title	varchar(255)
section_title	VARCHAR(255)

Event	
id	INT
event_name	varchar(255)
date	varchar(255)
time	varchar(255)
location	varchar(255)
EVENT SPECIFIC COLUMNS	varchar(255)

Version 4 DB

Sawyer Fenwick

Scraping

"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."
(http://agilemanifesto.org/principles.html, 2001)

Incorrect assumptions concerning team names

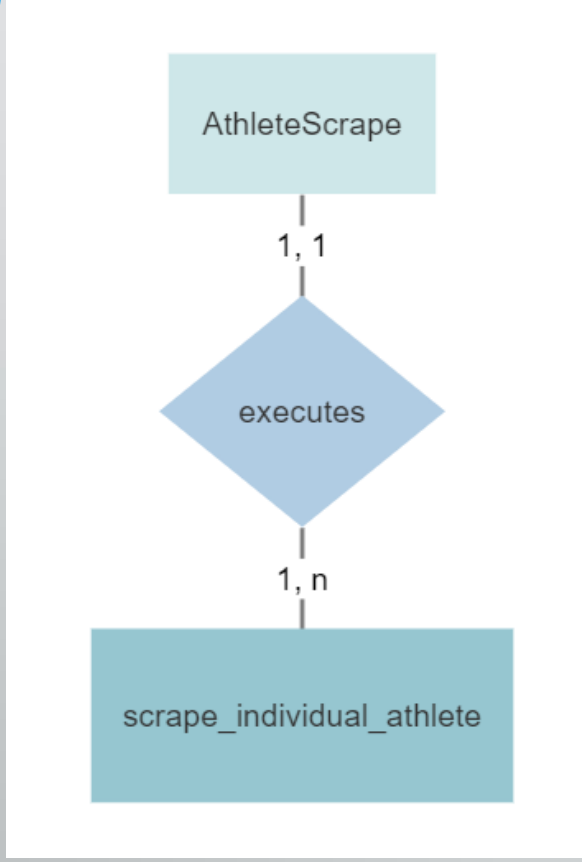
- Cheng/He
- Cheng/Li
- Cheng/Ngai
- Cheng/Zhang

Name	Event
AB	Hockey
AB	Artistic Swimming
AB	Wheelchair Basketball
AB	Speed Skating
AB	Gymnastics
AB	Table Tennis
AB	Table Tennis
AB	Biathlon
AB	Curling

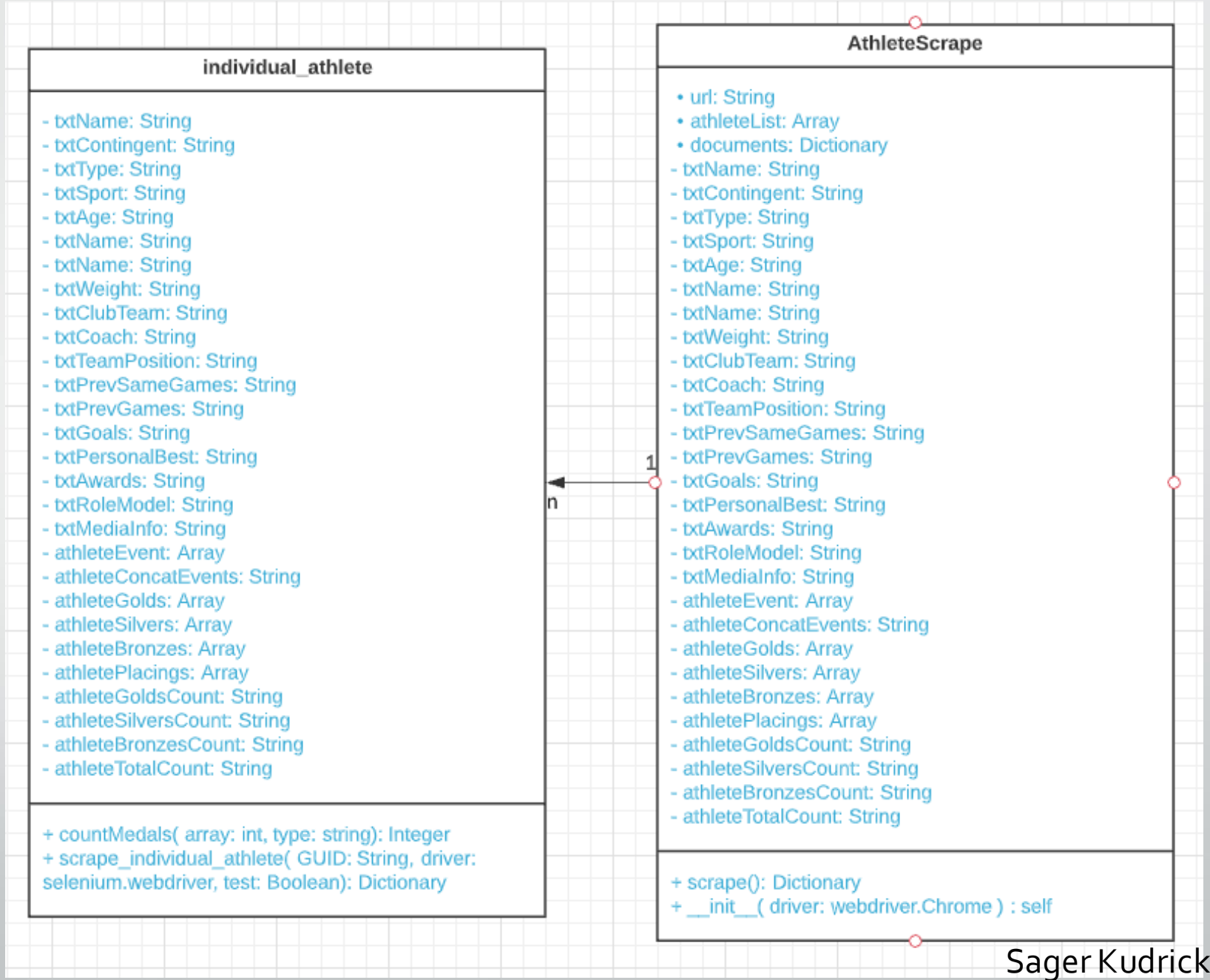
Sager Kudrick

Athletes

ERD



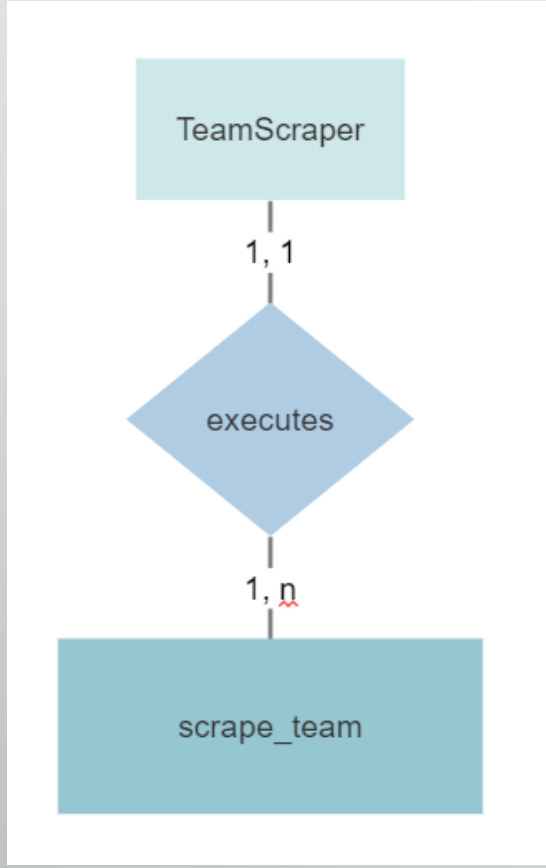
UML



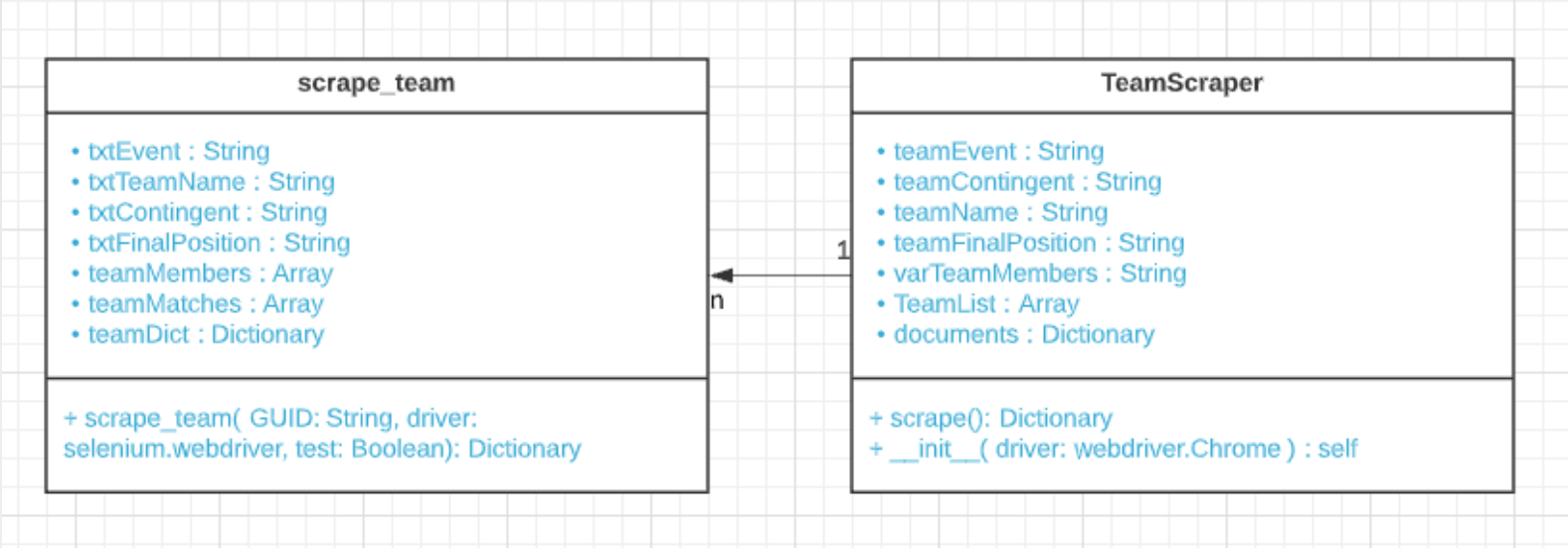
Sager Kudrick

Teams

ERD



UML



"Simplicity--the art of maximizing the amount of work not done--is essential."
(http://agilemanifesto.org/principles.html, 2001)

Sager Kudrick

Canada Games Events

NIAGARA 2022 JEUX DU CANADA GAMES

Niagara 2022 Canada Summer Games

EN | FR

Welcome | Sports | Schedule | Registration | Login

Sports

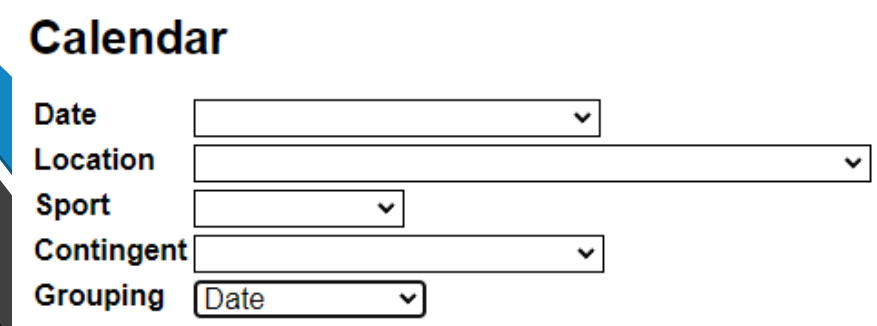
Sport	Sat 06/08	Sun 07/08	Mon 08/08	Tue 09/08	Wed 10/08	Thu 11/08	Fri 12/08	Sat 13/08	Sun 14/08	Mon 15/08	Tue 16/08	Wed 17/08	Thu 18/08	Fri 19/08	Sat 20/08	Sun 21/08
Athletics											●	●		●	●	
Baseball		●	●	●	●		●	●								
Basketball			●	●	●	●	●	●								
Box Lacrosse		●	●	●	●	●	●				●	●	●	●	●	●
Canoe Kayak											●	●	●	●		
Cycling			●		●		●				●		●		●	
Diving											●	●	●		●	●
Golf												●	●	●	●	
Rowing												●	●		●	●
Rugby Sevens			●	●												
Sailing												●	●	●	●	

©2022 KIMIK iT
All rights reserved
Privacy Policy Statement

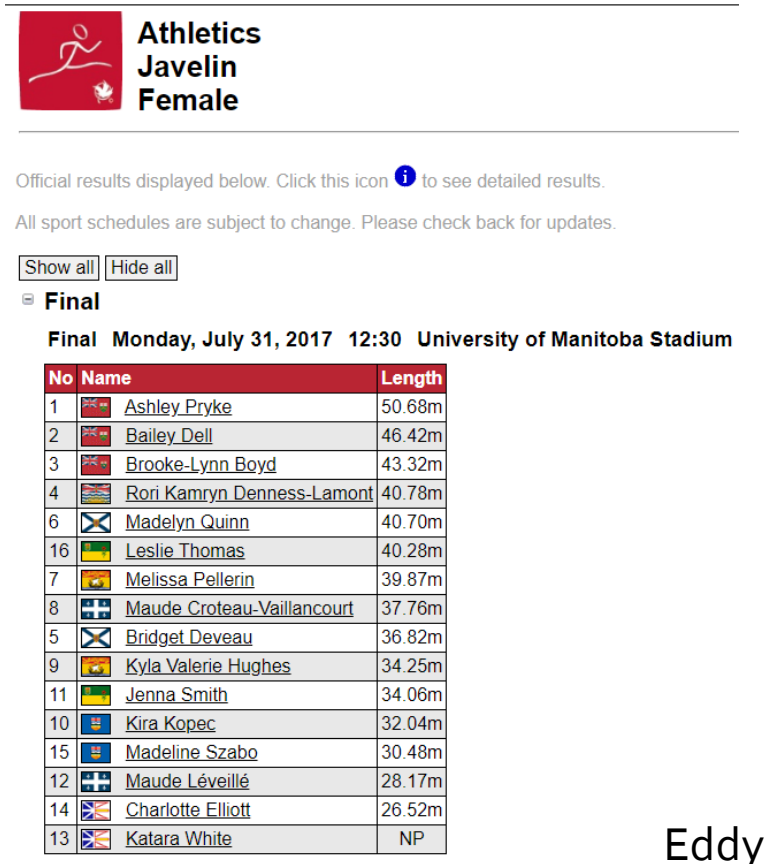
Eddy Su

Canada Games Events

Searching for Events



Individual Event Page



Athletics
Javelin
Female

Official results displayed below. Click this icon to see detailed results.

All sport schedules are subject to change. Please check back for updates.

Show all | Hide all

Final

Final Monday, July 31, 2017 12:30 University of Manitoba Stadium

No	Name	Length
1	Ashley Pryke	50.68m
2	Bailey Dell	46.42m
3	Brooke-Lynn Boyd	43.32m
4	Rori Kamryn Denness-Lamont	40.78m
6	Madelyn Quinn	40.70m
16	Leslie Thomas	40.28m
7	Melissa Pellerin	39.87m
8	Maude Croteau-Vaillancourt	37.76m
5	Bridget Deveau	36.82m
9	Kyla Valerie Hughes	34.25m
11	Jenna Smith	34.06m
10	Kira Kopec	32.04m
15	Madeline Szabo	30.48m
12	Maude Léveillé	28.17m
14	Charlotte Elliott	26.52m
13	Katara White	NP

Eddy Su

AI

Models

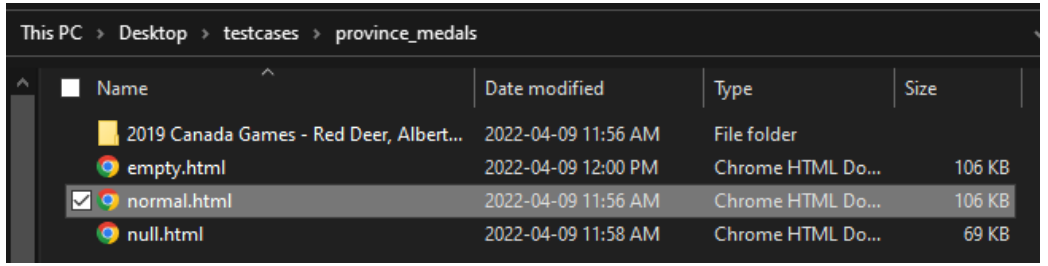
- Seq2Seq: Supervised learning model. Learning over a series of questions and answers.
- Seq2Sql: Weakly supervised model. Converts text to sql. Low accuracy.
- Google Tapas: Weakly supervised model. Seq2Seq answer. Takes a column header and list of tuples
- Haystack Framework: Weakly supervised model. Takes additional meta data such as table name and info.

Testing

Test Results

- Pass:
  - Unit level: The class method has returned the expected result.
  - Component level: All unit tests pass.
- Fail:
  - Unit level:
    - Exception raised.
    - The method does not return the expected output.
  - Component fail:
    - Any unit test failing.
    - Component requests rejected.
  - System level:
    - Not responding appropriately to user/admin requests.

Testing Province Medals



Medal standings

Contingent	Gold	Silver	Bronze	Total
Quebec	65	41	40	146
Ontario	18	43	44	105
Alberta	36	33	31	100
British Columbia	30	28	29	87
Manitoba	9	7	9	25
Saskatchewan	3	3	11	17
Nova Scotia	1	6	4	11
New Brunswick	1	3	5	9
Newfoundland and Labrador	1	0	1	2
Prince Edward Island	0	1	1	2
Northwest Territories	0	0	1	1
Yukon	0	0	1	1
Nunavut	0	0	0	0

Medal standings

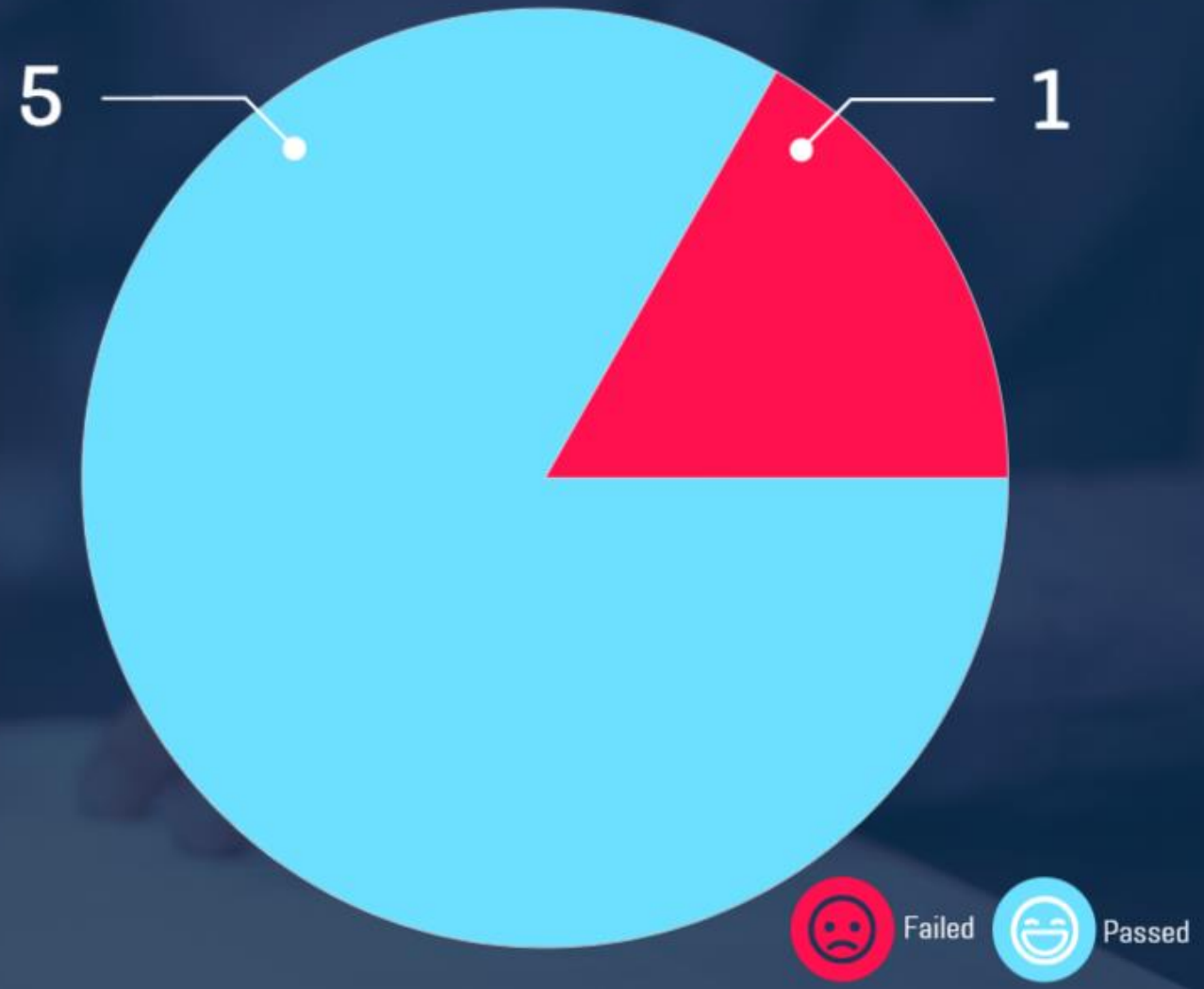
Medal standings

Contingent	Gold	Silver	Bronze	Total
Quebec	65	41	40	146
Ontario				
Alberta				
British Columbia				
Manitoba				
Saskatchewan				
Nova Scotia				
New Brunswick				
Newfoundland and Labrador				
Prince Edward Island				
Northwest Territories	0	0	1	1
Yukon	0	0	1	1
Nunavut	0	0	0	0

Sager Kudrick

Province Medals

2022-04-14 TESTS




```
Traceback (most recent call last):
  File "c:\Users\me\Desktop\Cosc4p02ChatbotProject-main\Chatbot\Back-End\components\executetests.py", line 30, in <module>
    provtest2 = prov2.scrape("file:///C:/users/me/desktop/testcases/province_medals/null.html", True)
  File "c:\Users\me\Desktop\Cosc4p02ChatbotProject-main\Chatbot\Back-End\components\scraping\modules\province_medals.py", line 80, in scrape
    'Province Name': provinceName,
UnboundLocalError: local variable 'provinceName' referenced before assignment
```

Sager Kudrick

Testing Continued

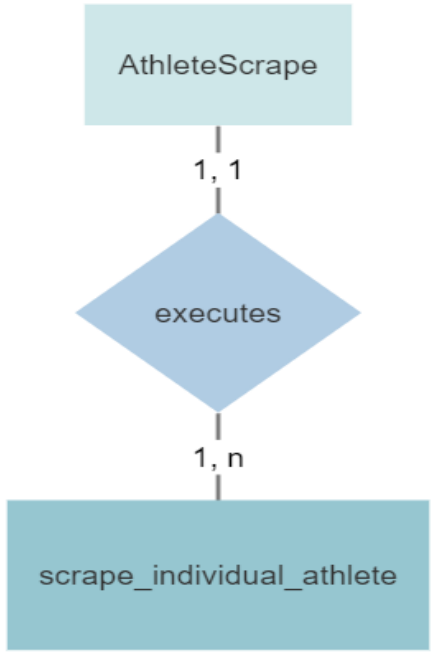
Unit testing

Unit testing is the process of **testing individual components in isolation**. It is a defect testing process. Units may be:

- **Individual functions** or methods within an object;
- **Object classes** with several attributes and methods;
- **Composite components** with defined interfaces used to access their functionality.

- **Unit testing**: individual program units or object classes are tested; should focus on testing the functionality of objects or methods.
- **Component testing**: several individual units are integrated to create composite components; should focus on testing component interfaces.
- **System testing**: some or all of the components in a system are integrated and the system is tested as a whole; should focus on testing component interactions.

Reference: Sommerville, Software Engineering, 10 ed., Chapter 8



Sager Kudrick

Conclusion