

CIS*3760: Chatron - Milestone 2 Summary

Note: server may not be running at the moment, so trying to test the Slack bot or website interface, <https://chatron.socs.uoguelph.ca/>, may not work.

User Stories

NOTE: We conducted a formal code review meeting where the team collectively examined our codebase, identifying areas of improvement as well as facilitating knowledge across the team.

Code Review Meeting - Nov 05 - 8:00pm

- Seegal presented an overview of current functionality (implementation details) of the bot at a low level to ensure the team is all on the same page and allow us to ask any questions for clarification as well as suggestions for improvements besides merge reviews
- Seegal goes over Wit.ai model training
 - Need slave labour to train Wit AI with lots of cases
 - How to train Wit AI, create intents, entities, etc
- Using nginx now so offline stuff doesn't matter
- npm start or npm test
 - Running bot vs test suite run
 - Demonstrated both to us, and talked about the code being run
- Goes over how he loaded data into local DB using python script from API calls over time
- Result of code review incorporated into his open merge request, will review again on GitLab due to merging process for approval before merging

User Story #1

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/6>

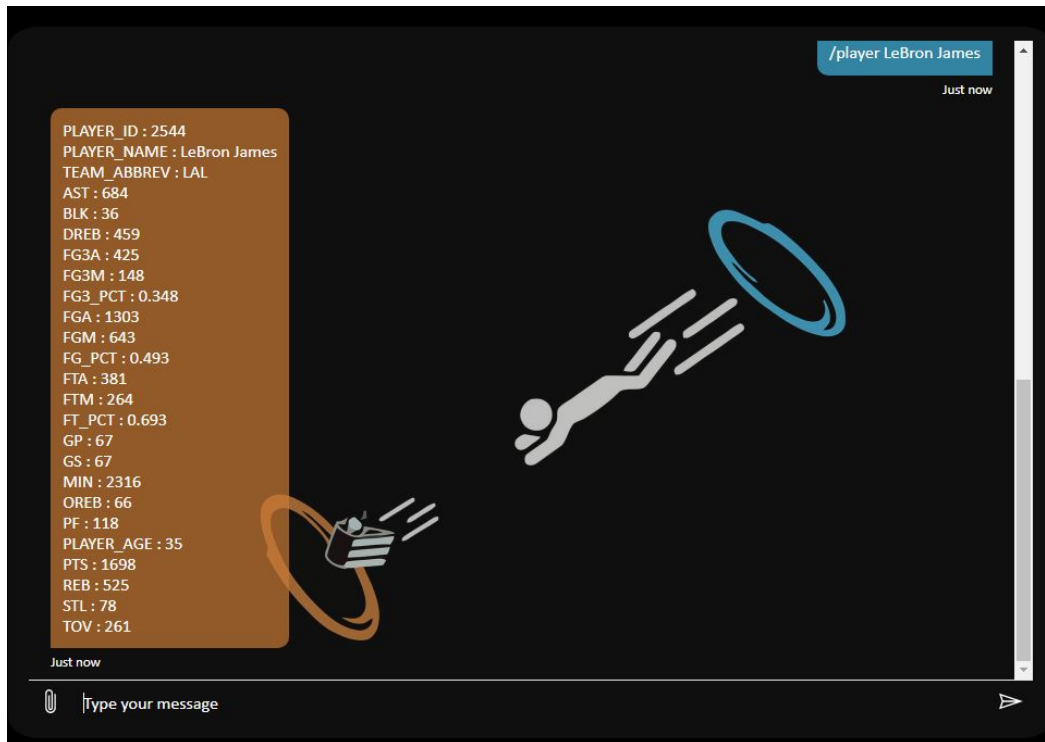
No epics, Proper Format, Acceptance Criteria

- It is a properly formatted user story: "As a <role>, I want [...] so that [...]"
- It is a broken-down task that accomplishes a single thing, i.e. not an epic
 - Part of "Epic: Bot Inquiries" (dark blue label)
- It has properly formatted acceptance criteria: "Given ... When ... Then ..."
 - As well as additional notes on desired outcome of the task
- Note there is also has a weight of 5, representing estimate of effort and time necessary

- Story also relates to #18 (<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/18>) because this story is to be completed using /command functionality using a player name. Later, story #21 (<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/21>), which relates to #20 (<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/20>), will also use this story because it will complete natural language processing functionality related to recognizing a player name.

Deliverable

- See user story acceptance criteria:
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/6> and definition of done
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/TeamStandards#issue-board-flow>
 - The client approved and accepted the delivery of basic functionality: they were able to see/use the product and receive responses to messages.
- See video in the following link under Milestone 2, User Story 1:
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/UserStoryDemoVideos#user-story-1>



Repeatable Testing

- See user story repeatable testing:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/6>

- All 3 of the automated test cases implemented at the time passed when merging into master:

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/22517>

```
31 > nba-chatron@1.0.0 test /builds/3760f20/chatron/wiki-chatbot
32 > node ./tests/bot.test.js
33 Test 1 passed.
34 Test 2 passed.
35 Test 3 passed.
```

- Automated test cases were *bot.test.js*;

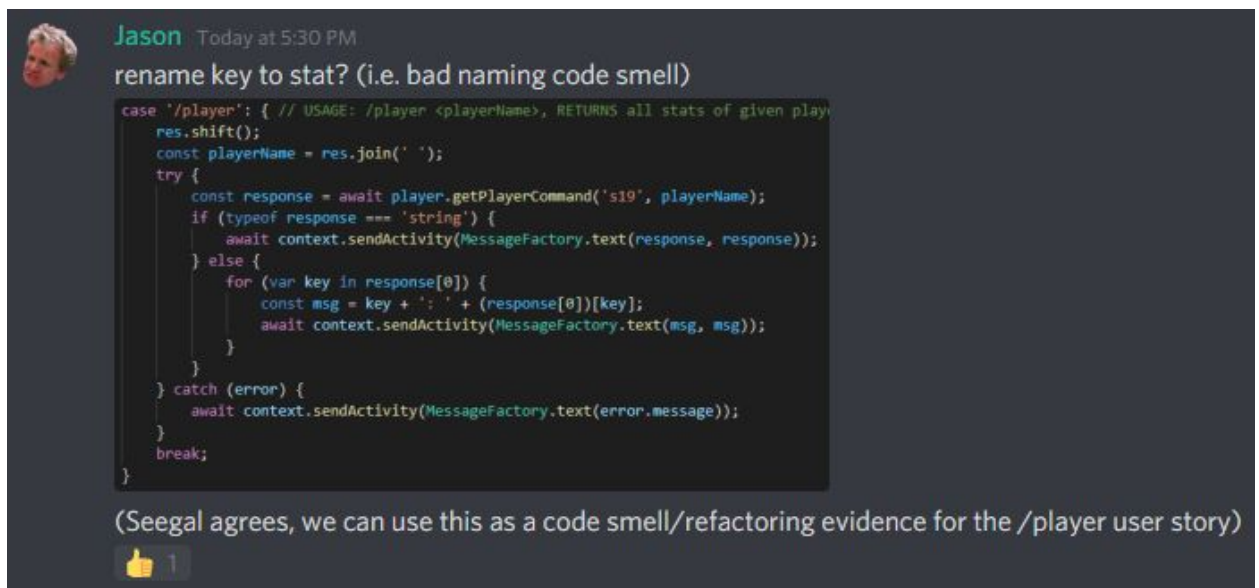
- <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/ac732e8aa44aa7b9a5bb7b97381d740e9e9cb1b6/tests/bot.test.js>

- In the latest pipeline testing job, functionality for this user story was tested on lines 34, 58, 60, 66, 69, 92, and 93 of the output:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23881>

Refactored

- Jason noticed a code smell for a variable name “key” (bad naming choice) and suggested a small refactor to rename it to something more descriptive (“stat”). In merge request 9 (https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/9), Seegal fixed the code smell in *command.js* by implementing the suggestion (among other refactors).
- Jason identifying the code smell and suggesting a fix:



- Previous code with the code smell (variable 'key' is not a descriptive name):

```
// Look at the command entered, and get the stat desired
switch (command) {
case '/player': { // USAGE: /player <playerName>, RETURNS all stats of given player
  res.shift();
  const playerName = res.join(' ');
  const response = await player.getPlayerCommand('s19', playerName);
  if (typeof response === 'string') {
    await context.sendActivity(MessageFactory.text(response, response));
  } else {
    for (var key in response[0]) {
      const msg = key + ': ' + (response[0])[key];
      await context.sendActivity(MessageFactory.text(msg, msg));
    }
  }
  break;
}
}
```

- <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/56fb9ff0e5ede02bb8978b37dbf07ff0210e5613/bot.js>

- As you can see in the below image, it now checks **stat** in response, rather than **key** in response:

```
async getPlayerCommand(playerName) {
  return new Promise((resolve, reject) => {
    const player = new Player();
    player.getPlayerStats('s19', playerName)
      .then((response) => {
        if (typeof response === 'string') return resolve(response);
        else {
          let msg = '';
          for (var stat in response[0]) {
            msg += `${stat} : ${ (response[0])[stat] } \n`;
          }
          return resolve(msg);
        }
      })
      .catch((error) => resolve(error.message));
  });
}
```

- <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/e2c58e6bfd787b00c4d3083d0b950651d9341be4/impl/command.js>
- Additionally, in the above image, the code was moved from *bot.js* into *command.js*, allowing for our bot code to simply take a message, hand it off to the other modules, and wait for them to generate a response (which it returns), thus removing bloat.

User Story #2

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/12>

No epics, Proper Format, Acceptance Criteria

- It is a properly formatted user story: “As a <role>, I want [...] so that [...]”
- It is a broken-down task that accomplishes a single thing, i.e. not an epic
 - Part of “Epic: Bot Inquiries” (dark blue label)
- It has properly formatted acceptance criteria: “Given ... When ... Then ...”
 - As well as additional notes on desired outcome of the task
- Note there is also has a weight of 5, representing estimate of effort and time necessary
- Story also relates to #18

(<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/18>) because this story is to be completed using /command functionality using a player name and stat.

Later, story #21

(<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/21>) and story #23

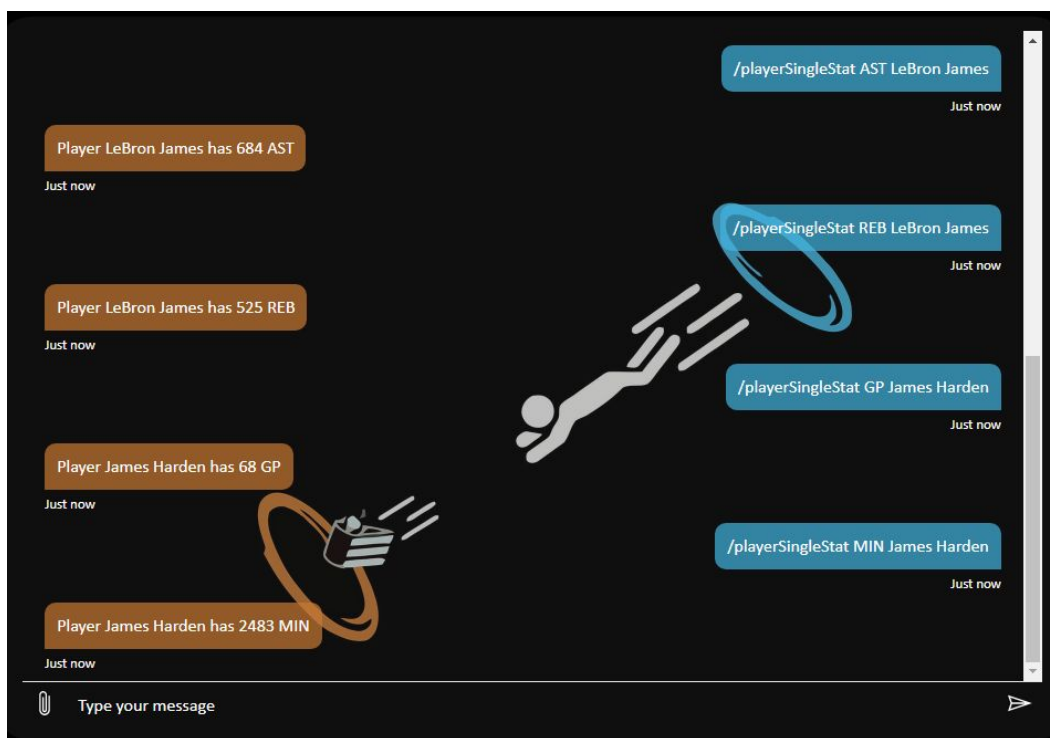
(<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/23>), where both

relate to #20 (<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/20>),

will also use this story because it will complete natural language processing functionality related to recognizing a player name and stat.

Deliverable

- See user story acceptance criteria:
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/12> and definition of done
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/TeamStandards#issue-board-flow>
 - The client approved and accepted the delivery of basic functionality: they were able to see/use the product and receive responses to messages.
- See video in the following link under Milestone 2, User Story 2:
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/UserStoryDemoVideos#user-story-2>



- Etc. for more stats and more players

Repeatable Testing

- See user story repeatable testing:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/12>

- All of the 7 automated test cases implemented at the time passed when merging into master:

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23563>

```
31 > nba-chatron@1.0.0 test /builds/3760f20/chatron/wiki-chatbot
32 > nyc mocha
33 Database
34   Find player Kim Kardashian
35     ✓ should respond with zero rows found (141ms)
36   Find player LeBron James
37     ✓ should respond with an object of length > 0
38     ✓ should respond with LeBron James' stats
39 Player
40   Find player LeBron James
41     ✓ should respond with an object
42     ✓ should respond with LeBron James
43 Wit
44   Analyze a conversational question for a single player stat
45     ✓ should detect the intent is playerSingleStat (249ms)
46     ✓ should detect the name of the player (218ms)
47 7 passing (646ms)
```

- Automated test cases were *player.test.js*, *database.test.js*, and *wit.test.js*;
 - <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/9aad11a9df2da6641be48629a779368edde3a8f5/test/player.test.js>
 - <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/9aad11a9df2da6641be48629a779368edde3a8f5/test/database.test.js>
 - <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/9aad11a9df2da6641be48629a779368edde3a8f5/test/wit.test.js>

- In the latest pipeline testing job, functionality for this user story was tested on lines 34, 36, 39, 63, 72, 75, 92, and 97 of the output:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23881>

Refactored

- We noticed that we were placing all the command handling functionality in a single function, which made it grow much larger with each command we added. This was identified as the ‘large function/method’ code smell. In merge request 9 (https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/9), Seegal abstracted the command parsing functionality out of *bot.js* and created a separate class in *command.js* called ‘Command’. There, each command had its own small function to handle the command.
- Too much code to post screenshots here. See the following link and observe the changes in *bot.js* and *command.js*:
https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/9/diffs.
Notice how a large amount of code was removed from the async arrow function in *bot.js* and replaced in *commands.js*.

User Story #3

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/13>

No epics, Proper Format, Acceptance Criteria

- It is a properly formatted user story: “As a <role>, I want [...] so that [...]”
- It is a broken-down task that accomplishes a single thing, i.e. not an epic
 - Part of “Epic: Bot Inquiries” (dark blue label)
- It has properly formatted acceptance criteria: “Given ... When ... Then ...”
 - As well as additional notes on desired outcome of the task
- Note there is also has a weight of 5, representing estimate of effort and time necessary
- Story also relates to #18

(<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/18>) because this

story is to be completed using /command functionality using a stat. This story is very similar and has a twin story for a “lowest” version, story #35

(<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/35>). Later, story #23 (<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/23>) and story #38 (<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/38>), where both relate to #20

(<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/20>), will also use this story because it will complete natural language processing functionality related to recognizing a stat and an order ranking qualifier.

Deliverable

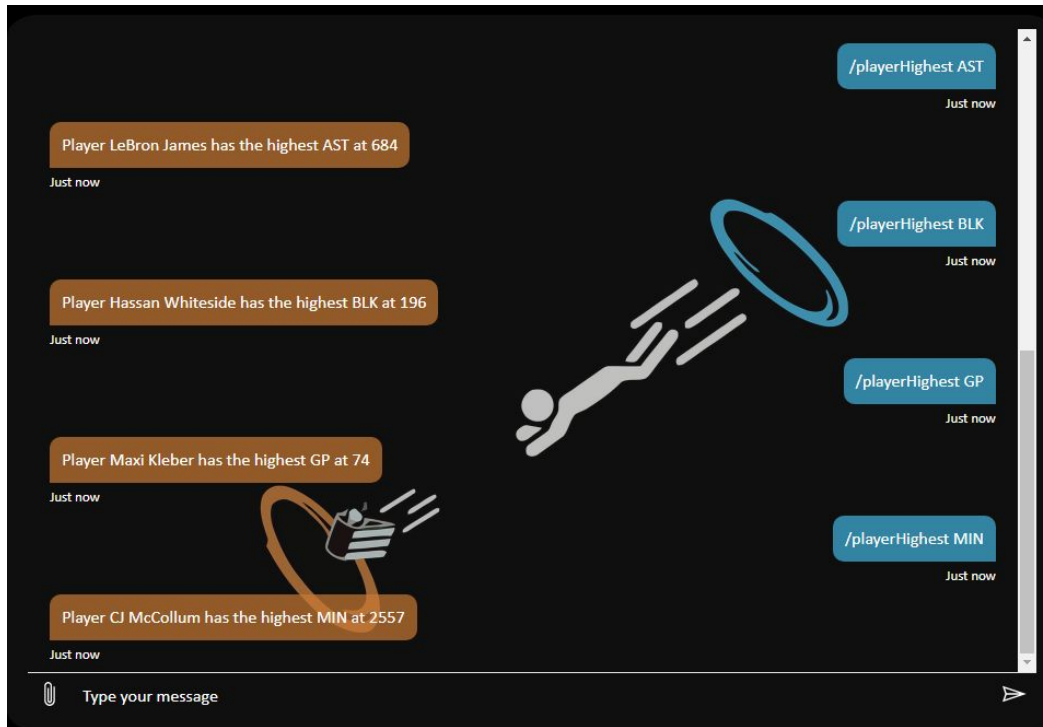
- See user story acceptance criteria:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/13> and definition of done

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/TeamStandards#issue-board-flow>

- The client approved and accepted the delivery of basic functionality: they were able to see/use the product and receive responses to messages.
- See video in the following link under Milestone 2, User story 3:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/UserStoryDemoVideos#user-story-3>



- Etc. for more stats

Repeatable Testing

- See user story repeatable testing:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/issues/13>

- All of the 7 automated test cases implemented at the time passed when merging into master:

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23579>

```
31 > nba-chatron@1.0.0 test /builds/3760f20/chatron/wiki-chatbot
32 > nyc mocha
33 Database
34   Find player Kim Kardashian
35     ✓ should respond with zero rows found (116ms)
36   Find player LeBron James
37     ✓ should respond with an object of length > 0
38     ✓ should respond with LeBron James' stats
39 Player
40   Find player LeBron James
41     ✓ should respond with an object
42     ✓ should respond with LeBron James
43 Wit
44   Analyze a conversational question for a single player stat
45     ✓ should detect the intent is playerSingleStat (269ms)
46     ✓ should detect the name of the player (174ms)
47 7 passing (603ms)
```

- Automated test cases were *player.test.js*, *database.test.js*, and *wit.test.js*;

- <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/345c4f74b3a14aa8f5ba6cb2513cd5ccf8291da9/test/player.test.js>
- <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/345c4f74b3a14aa8f5ba6cb2513cd5ccf8291da9/test/database.test.js>
- <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/345c4f74b3a14aa8f5ba6cb2513cd5ccf8291da9/test/wit.test.js>

- In the latest pipeline testing job, functionality for this user story was tested on lines 34, 41, 44, 46, 49, 63, 78, 82, 85, 89, and 92 of the output:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23881>

Refactored

- We identified duplicate code in the code for */playerLowest* and */playerHighest*. In merge request 9

(https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/9), Seegal

addressed this by combining the two into a single function.

- Duplicate code:

```
async getHighestCommand(table, stat) {
  return new Promise((resolve, reject) => {
    const db = new DBConnection();
    const query = 'SELECT PLAYER_NAME, ' + stat + ' FROM ' + table + ' ORDER BY ' + stat + ' DESC;';
    db.query(query)
      .then((response) => resolve(response))
      .catch(() => reject(Error('Stat passed is invalid.')));
  });
}

async getLowestCommand(table, stat) {
  return new Promise((resolve, reject) => {
    const db = new DBConnection();
    const query = 'SELECT PLAYER_NAME, ' + stat + ' FROM ' + table + ' ORDER BY ' + stat + ' ASC;';
    db.query(query)
      .then((response) => resolve(response))
      .catch(() => reject(Error('Stat passed is invalid.')));
  });
}
```

- <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/fba2ec58172581b0ce98ef4d0e3632b86ed63c80/impl/player.js>

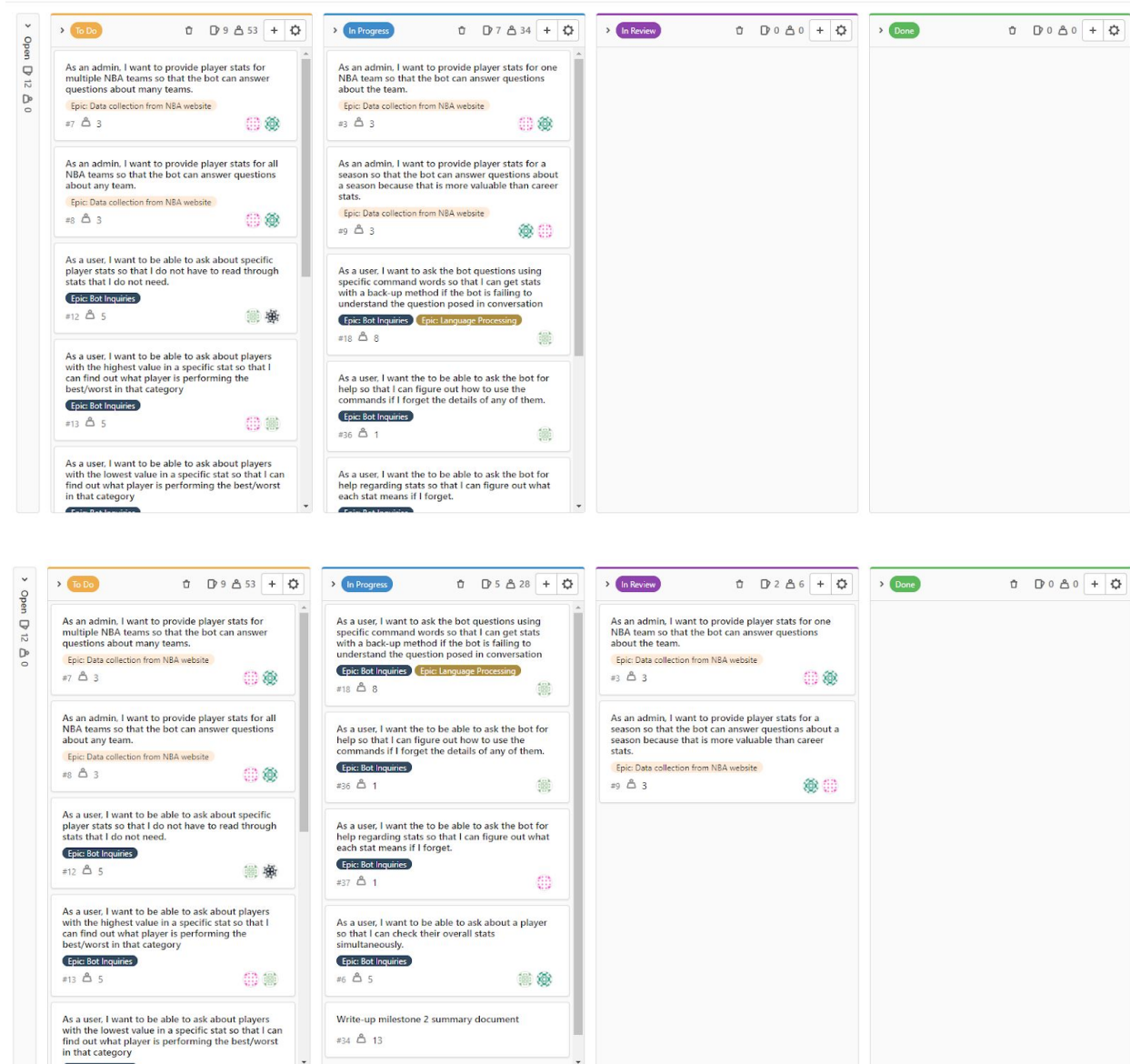
- Refactored code:

```
async getStatHighestLowestPlayer(table, stat, order) {
  return new Promise((resolve, reject) => {
    const db = new DBConnection();
    const query = `SELECT PLAYER_NAME, ${stat} FROM ${table} ORDER BY ${stat} ${order}`;
    db.query(query)
      .then((response) => resolve(response))
      .catch(() => reject(Error('Stat passed is invalid.')));
  });
}
```

- <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/e2c58e6bfd787b00c4d3083d0b950651d9341be4/impl/player.js>

Scrum Board Screenshots of Progress Over Time

- Top is oldest screenshot, bottom is most recent - taken periodically over the sprint



To Do

850

As an admin, I want to provide player stats for all NBA teams so that the bot can answer questions about any team.

Epic: Data collection from NBA website

#8 3

As a user, I want to be able to ask about specific player stats so that I do not have to read through stats that I do not need.

Epic: Bot Inquiries

#12 5

As a user, I want to be able to ask about players with the highest value in a specific stat so that I can find out what player is performing the best/worst in that category

Epic: Bot Inquiries

#13 5

As a user, I want to be able to ask about players with the lowest value in a specific stat so that I can find out what player is performing the best/worst in that category

Epic: Bot Inquiries

#35 5

As a user, I want to be able to talk to the bot as naturally as possible, like I'm having a conversation so that I don't need to memorize various commands for the bot.

In Progress

321

As a user, I want to be able to ask about a player so that I can check their overall stats simultaneously.

Epic: Bot Inquiries

#6 5

As an admin, I want to provide player stats for multiple NBA teams so that the bot can answer questions about many teams.

Epic: Data collection from NBA website

#7 3

Write-up milestone 2 summary document

#34 13

In Review

310

As a user, I want to ask the bot questions using specific command words so that I can get stats with a back-up method if the bot is failing to understand the question posed in conversation

Epic: Bot InquiriesEpic: Language Processing

#18 8

As a user, I want the to be able to ask the bot for help so that I can figure out how to use the commands if I forget the details of any of them.

Epic: Bot Inquiries

#36 1

As a user, I want the to be able to ask the bot for help regarding stats so that I can figure out what each stat means if I forget.

Epic: Bot Inquiries

#37 1

Done

26

As an admin, I want to provide player stats for one NBA team so that the bot can answer questions about the team.

Epic: Data collection from NBA website

#3 3

As an admin, I want to provide player stats for a season so that the bot can answer questions about a season because that is more valuable than career stats.

Epic: Data collection from NBA website

#9 3

To Do

642

As a user, I want to be able to ask about players with the highest value in a specific stat so that I can find out what player is performing the best/worst in that category

Epic: Bot Inquiries

#13 5

As a user, I want to be able to ask about players with the lowest value in a specific stat so that I can find out what player is performing the best/worst in that category

Epic: Bot Inquiries

#35 5

As a user, I want to be able to talk to the bot as naturally as possible, like I'm having a conversation so that I don't need to memorize various commands for the bot.

Epic: Bot InquiriesEpic: Language Processing

#20 13

As a user, I want the bot to tell me it doesn't understand my inquiry if it cannot process it so that I can reword my inquiry as needed.

Epic: Language Processing

#24 3

As a user, I want the bot to understand what specific player I'm asking about so that I can inquire about specific stats or info about the

In Progress

321

As an admin, I want to provide player stats for all NBA teams so that the bot can answer questions about any team.

Epic: Data collection from NBA website

#8 3

As a user, I want to be able to ask about specific player stats so that I do not have to read through stats that I do not need.

Epic: Bot Inquiries

#12 5

Write-up milestone 2 summary document

#34 13

In Review

28

As a user, I want to be able to ask about a player so that I can check their overall stats simultaneously.

Epic: Bot Inquiries

#6 5

As an admin, I want to provide player stats for multiple NBA teams so that the bot can answer questions about many teams.

Epic: Data collection from NBA website

#7 3

Done

516

As a user, I want the to be able to ask the bot for help regarding stats so that I can figure out what each stat means if I forget.

Epic: Bot Inquiries

#37 1

As a user, I want the to be able to ask the bot for help so that I can figure out how to use the commands if I forget the details of any of them.

Epic: Bot Inquiries

#36 1

As a user, I want to ask the bot questions using specific command words so that I can get stats with a back-up method if the bot is failing to understand the question posed in conversation

Epic: Bot InquiriesEpic: Language Processing

#18 8

As an admin, I want to provide player stats for a season so that the bot can answer questions about a season because that is more valuable than career stats.

Epic: Data collection from NBA website

#9 3

As an admin, I want to provide player stats for one NBA team so that the bot can answer questions about the team.

Epic: Data collection from NBA website

To Do

319

As a user, I want the bot to tell me it doesn't understand my inquiry if it cannot process it so that I can reword my inquiry as needed.

Epic: Language Processing

#243

As a user, I want the bot to understand what specific player I'm asking about so that I can inquire about specific stats or info about the player.

Epic: Language Processing

#218

As a user, I want the bot to understand what specific statistic I'm asking about so that I can inquire about specifics of the stat.

Epic: Language Processing

#238

In Progress

436

As a user, I want to be able to ask about players with the highest value in a specific stat so that I can find out what player is performing the best/worst in that category

Epic: Bot Inquiries

#135

As a user, I want to be able to ask about players with the lowest value in a specific stat so that I can find out what player is performing the best/worst in that category

Epic: Bot Inquiries

#355

As a user, I want to be able to talk to the bot as naturally as possible, like I'm having a conversation so that I don't need to memorize various commands for the bot.

Epic: Bot InquiriesEpic: Language Processing

#2013

Write-up milestone 2 summary document

#3413

In Review

28

As an admin, I want to provide player stats for all NBA teams so that the bot can answer questions about any team.

Epic: Data collection from NBA website

#83

As a user, I want to be able to ask about specific player stats so that I do not have to read through stats that I do not need.

Epic: Bot Inquiries

#125

Done

724

As an admin, I want to provide player stats for multiple NBA teams so that the bot can answer questions about many teams.

Epic: Data collection from NBA website

#73

As a user, I want to be able to ask about a player so that I can check their overall stats simultaneously.

Epic: Bot Inquiries

#65

As a user, I want the to be able to ask the bot for help regarding stats so that I can figure out what each stat means if I forget.

Epic: Bot Inquiries

#371

As a user, I want the to be able to ask the bot for help so that I can figure out how to use the commands if I forget the details of any of them.

Epic: Bot Inquiries

#361

As a user, I want to ask the bot questions using specific command words so that I can get stats with a back-up method if the bot is failing to understand the question posed in conversation

Epic: Bot InquiriesEpic: Language Processing

To Do

319

As a user, I want the bot to tell me it doesn't understand my inquiry if it cannot process it so that I can reword my inquiry as needed.

Epic: Language Processing

#243

As a user, I want the bot to understand what specific player I'm asking about so that I can inquire about specific stats or info about the player.

Epic: Language Processing

#218

As a user, I want the bot to understand what specific statistic I'm asking about so that I can inquire about specifics of the stat.

Epic: Language Processing

#238

In Progress

218

As a user, I want to be able to ask about players with the lowest value in a specific stat so that I can find out what player is performing the best/worst in that category

Epic: Bot Inquiries

#355

As a user, I want to be able to talk to the bot as naturally as possible, like I'm having a conversation so that I don't need to memorize various commands for the bot.

Epic: Bot InquiriesEpic: Language Processing

#2013

In Review

218

As a user, I want to be able to ask about players with the highest value in a specific stat so that I can find out what player is performing the best/worst in that category

Epic: Bot Inquiries

#135

Write-up milestone 2 summary document

#3413

Done

932

As a user, I want to be able to ask about specific player stats so that I do not have to read through stats that I do not need.

Epic: Bot Inquiries

#125

As an admin, I want to provide player stats for all NBA teams so that the bot can answer questions about any team.

Epic: Data collection from NBA website

#83

As an admin, I want to provide player stats for multiple NBA teams so that the bot can answer questions about many teams.

Epic: Data collection from NBA website

#73

As a user, I want to be able to ask about a player so that I can check their overall stats simultaneously.

Epic: Bot Inquiries

#65

As a user, I want the to be able to ask the bot for help regarding stats so that I can figure out what each stat means if I forget.

Epic: Bot Inquiries

#371

The image displays a Jira board with two rows of four columns each, representing different stages of a project: To Do, In Progress, In Review, and Done. Each column contains several user stories or tasks with their IDs, priorities, and status labels.

Row 1:

- To Do:**
 - #24: As a user, I want the bot to tell me it doesn't understand my inquiry if it cannot process it so that I can reword my inquiry as needed. (Epic: Language Processing)
 - #21: As a user, I want the bot to understand what specific player I'm asking about so that I can inquire about specific stats or info about the player. (Epic: Language Processing)
 - #23: As a user, I want the bot to understand what specific statistic I'm asking about so that I can inquire about specifics of the stat. (Epic: Language Processing)
 - #38: As a user, I want the bot to understand what specific order ranking I'm asking about so that I can inquire about the player doing the best/worst in a stat category. (Epic: Language Processing)
- In Progress:**
 - #20: As a user, I want to be able to talk to the bot as naturally as possible, like I'm having a conversation so that I don't need to memorize various commands for the bot. (Epic: Bot Inquiries, Epic: Language Processing)
- In Review:**
 - #35: As a user, I want to be able to ask about players with the lowest value in a specific stat so that I can find out what player is performing the best/worst in that category. (Epic: Bot Inquiries)
 - #34: Write-up milestone 2 summary document
- Done:**
 - #13: As a user, I want to be able to ask about players with the highest value in a specific stat so that I can find out what player is performing the best/worst in that category. (Epic: Bot Inquiries)
 - #12: As a user, I want to be able to ask about specific player stats so that I do not have to read through stats that I do not need. (Epic: Bot Inquiries)
 - #8: As an admin, I want to provide player stats for all NBA teams so that the bot can answer questions about any team. (Epic: Data collection from NBA website)
 - #7: As an admin, I want to provide player stats for multiple NBA teams so that the bot can answer questions about many teams. (Epic: Data collection from NBA website)
 - #6: As a user, I want to be able to ask about a player so that I can check their overall stats simultaneously. (Epic: Bot Inquiries)

Row 2:

- To Do:**
 - #21: As a user, I want the bot to understand what specific player I'm asking about so that I can inquire about specific stats or info about the player. (Epic: Language Processing)
 - #23: As a user, I want the bot to understand what specific statistic I'm asking about so that I can inquire about specifics of the stat. (Epic: Language Processing)
 - #38: As a user, I want the bot to understand what specific order ranking I'm asking about so that I can inquire about the player doing the best/worst in a stat category. (Epic: Language Processing)
- In Progress:**
 - #20: As a user, I want to be able to talk to the bot as naturally as possible, like I'm having a conversation so that I don't need to memorize various commands for the bot. (Epic: Bot Inquiries, Epic: Language Processing)
 - #24: As a user, I want the bot to tell me it doesn't understand my inquiry if it cannot process it so that I can reword my inquiry as needed. (Epic: Language Processing)
- In Review:**
 - #34: Write-up milestone 2 summary document
- Done:**
 - #35: As a user, I want to be able to ask about players with the lowest value in a specific stat so that I can find out what player is performing the best/worst in that category. (Epic: Bot Inquiries)
 - #13: As a user, I want to be able to ask about players with the highest value in a specific stat so that I can find out what player is performing the best/worst in that category. (Epic: Bot Inquiries)
 - #12: As a user, I want to be able to ask about specific player stats so that I do not have to read through stats that I do not need. (Epic: Bot Inquiries)
 - #8: As an admin, I want to provide player stats for all NBA teams so that the bot can answer questions about any team. (Epic: Data collection from NBA website)
 - #7: As an admin, I want to provide player stats for multiple NBA teams so that the bot can answer questions about many teams. (Epic: Data collection from NBA website)

Note: of course, the write-up summary task is “Done” when this is submitted.

Optional Elements

NOTE: see Team Standards doc for a brief merge and testing discussion in #committing-code:
<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/TeamStandards>

1. Testing

1.1 Full Suite of Unit Tests

Our unit test suite consists of four test files containing 41 tests, which are automatically run as a part of the build pipeline while merging code to the master branch. The 4 test files are *player.test.js*, *database.test.js*, *command.test.js*, and *wit.test.js*:

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/master/test/player.test.js>

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/master/test/database.test.js>

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/master/test/command.test.js>

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/master/test/wit.test.js>

Please see below for information on the coverage report as well as descriptions of each test file. The coverage report and test results for the latest merge request can be found at the following link: <https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23881>

1.2 Coverage Report

41 passing (801ms)					
File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	95.33	86.96	95.24	97.98	
command.js	92.19	80	90.91	96.67	31,121
database.js	100	100	100	100	
player.js	100	100	100	100	
wit.js	100	100	100	100	

As you can see above, our code coverage is fully functional and our team is proud. The only uncovered lines are due database system failures that are only caught if it fails to query the database, which will only occur if the database goes down or is offline. If you view the CI/CD pipeline on GitLab and view the third stage/phase, “testing”, for any pipeline, you should see the testing terminal output with coverage report at the end. Alternatively, you can view the “Jobs” part of CI/CD (i.e. history for all pipelines shown) and select a job number to view this (<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs>). The files tested all relate to functionality and are briefly explained below:

- **command.js** - tests for bot interaction functionality, covering each command using valid and invalid input. For example, some commands covered include “/player playerName”, “/playerSingleStat stat playerName”, “/playerHighest stat”, and “/playerLowest stat”. For each command, there are tests that will ensure proper functionality that only evaluate as successful if the output matches the expected output. In addition, some tests expect failure, such as when executing commands with non-existent players or bad input. This mainly tests the analysis/parsing of slash commands from a user message (eg. get the /command and any arguments), whereas tests for *player.js* are used to actually query the database for an answer. Note, for some of the tests for *command.js*, there begin to be integration tests because we validate a sample user command message and result by mocking it and hardcoding the expected answer.
- **database.js** - unit tests for bot query functionality since it queries the database for results about players and stats in order to answer questions from the user. In other words, we cover testing that tables exist, and columns that we search (for player name or a specific stat) also exist by attempting some success and failure cases. Of course, there’s tests for successful queries that don’t return any results since no data was found (eg. player name not in database).

- **player.js** - tests here are similar to that of those for *command.js*, except these already have “parsed” values from some user message. We use these to call a function which queries the database, so these are not unit tests, they are integration tests.
- **wit.js** - unit tests to confirm that Wit AI is functioning and able to properly recognize intent and entities for given messages, including being able to give proper responses when the intent of question by user is not recognized. At least we plan to add these latter tests once we have properly finished training the AI, since it currently needs more training before full testing and implementation into the product. In other words, the code for these stories haven’t been started yet but we plan to start them next sprint.

1.3 Integration Testing Plan

Our implementation of Web/Slack is different from other people in our class since we do not use the slack-adaptor node package. Our connection is conducted via Microsoft’s Direct Line API on Microsoft’s dev.botframework.com. All messages are routed through the Direct Line channel, thus full end-to-end testing is not feasible. Our plan for this involves 4 main features: Slack, WebChat, Wit.ai, and database; in reference to testing the bot:

- **Slack** - To test Slack, we will need to use Slack’s Web API, which will post messages to a channel then forwarded to the bot. At this point, we will retrieve any messages the bot returns. However, there is a major issue with this testing because the bot needs to be running for this test to pass. As such, we will need to set up an Auto Deployment as part of our GitLab CI/CD pipeline to test this feature.
- **WebChat** - Testing the webchat is more difficult than testing Slack. Due to the lack of an API, this will require us to use a web crawler in order to connect to our website and send a message via the text field, then parse any replies by the bot. Again, this has the same issue as Slack, where we will need to set up an auto-deployment feature as part of GitLab CI/CD pipeline, as the bot needs to be running/online to access the website.

- **Wit.ai** - As part of our current testing suite, we have unit tests for Wit.ai. Leading from this, we can introduce integration testing after we add natural language processing features to the product. The integration tests will be similar to our tests for *command.js* since they will send a message and see how it is handled at every step of the message-handling code.
- **Database** - Both database unit testing and integration testing is already part of our current testing suite: some tests in *command.js* and all in *player.js* currently query the database because there is no mock set up. As such, there is currently a mix of integration and unit testing. Essentially, our plan for database testing will be continuing the current testing implementation using Mocha, while introducing any new modules that will connect to the Database.

In general, our plan for integration testing does not extend to full end-to-end (E2E) integration testing. Currently, if the bot is running and the web server/slackbot is online, then sending a message to WebChat/Slack also works. Since we already have integration tests for some aspects, we just need to make sure we have integration tests for other pairs of functional units later so everything is covered by implication. For example, a very simple test could be where a message is sent in WebChat/Slack (doesn't matter what's sent) and it checks if something is returned (doesn't matter what's replied) because all that's necessary is to validate that the bot is correctly connected and can answer questions. However, more would be needed for end-to-end testing, but integration testing of webpage/Slack combined with our unit/integration test suite for our message-handling code should suffice.

1.4 Testing Tools Used and Justification of Choice

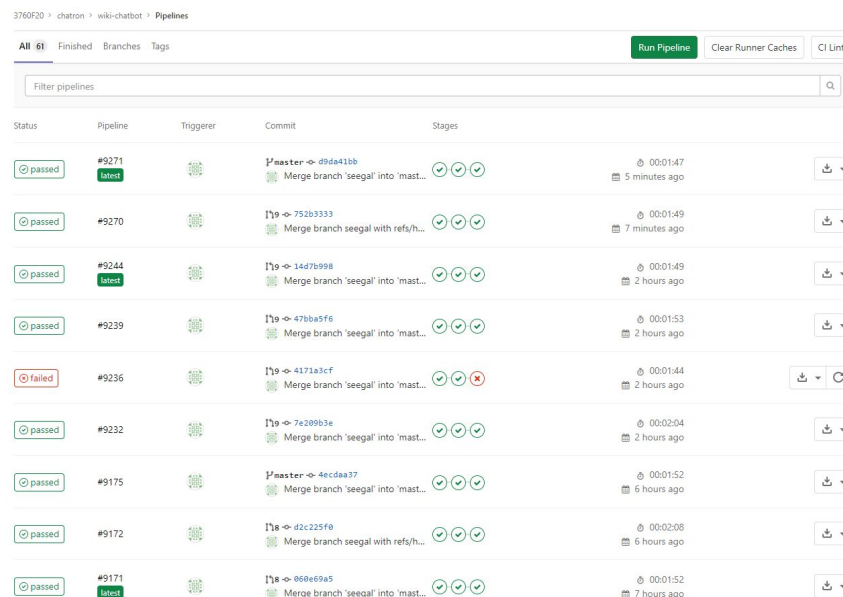
The testing tools our team used were **GitLab CI/CD pipeline**, **ESLint** as linter, **Mocha** for testing, and **Istanbul** for coverage reports. The GitLab CI/CD pipeline runs 3 jobs, which builds the code, checks linting using ESLint, and then runs tests using Mocha before finally

creating a coverage report using Istanbul. Our justifications for the testing tools chosen and used are below:

- **GitLab CI/CD Pipeline - Automatic Deployment**

- This is mostly self-explanatory since GitLab's CI/CD pipeline allows us to run the test on every commit for merge requests. Additionally, before any merges, GitLab does a temporary merge with master, and runs the test on this to prevent errors from propagating into production (the master branch).
- CI/CD Pipelines evidence:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/pipelines>



Status	Pipeline	Triggerer	Commit	Stages	Time
passed	#9271	latest	!master -> 69da41bb Merge branch 'seegal' into 'mast...	✓ ✓ ✓	00:01:47 5 minutes ago
passed	#9270	latest	!19 -> 752b3333 Merge branch 'seegal' with refs/h...	✓ ✓ ✓	00:01:49 7 minutes ago
passed	#9264	latest	!19 -> 14d7b998 Merge branch 'seegal' into 'mast...	✓ ✓ ✓	00:01:49 2 hours ago
passed	#9239	latest	!19 -> 47bba5f6 Merge branch 'seegal' into 'mast...	✓ ✓ ✓	00:01:53 2 hours ago
failed	#9236	latest	!19 -> 4171a3cf Merge branch 'seegal' into 'mast...	✓ ✓ ✗	00:01:44 2 hours ago
passed	#9232	latest	!19 -> 7e289b3e Merge branch 'seegal' into 'mast...	✓ ✓ ✓	00:02:04 2 hours ago
passed	#9175	latest	!master -> 4ecdaa37 Merge branch 'seegal' into 'mast...	✓ ✓ ✓	00:01:52 6 hours ago
passed	#9172	latest	!18 -> d2c225f8 Merge branch 'seegal' with refs/h...	✓ ✓ ✓	00:02:08 6 hours ago
passed	#9171	latest	!18 -> 869e69a5 Merge branch 'seegal' into 'mast...	✓ ✓ ✓	00:01:52 7 hours ago

(Notice that pipeline #9236 failed tests)

Or alternatively, CI/CD Pipeline Jobs evidence:

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs>

3760f20 > chatron > wiki-chatbot > Jobs

All 163Pending 0Running 0Finished 159

CI lint

Status	Job	Pipeline	Stage	Name	Timing	Coverage
<div>passed</div>	#23881 ✓ master → d9da41bb	#9271 by <div></div>	test	testing	<div>00:00:30</div> <div>8 minutes ago</div>	<div></div>
<div>passed</div>	#23880 ✓ master → d9da41bb	#9271 by <div></div>	lint	eslint	<div>00:00:36</div> <div>8 minutes ago</div>	<div></div>
<div>passed</div>	#23879 ✓ master → d9da41bb	#9271 by <div></div>	build	install_dependencies	<div>00:00:41</div> <div>9 minutes ago</div>	<div></div> <div></div>
<div>passed</div>	#23878 ✓ refs/merge-r... → 752b3333	#9270 by <div></div>	test	testing	<div>00:00:29</div> <div>10 minutes ago</div>	
<div>passed</div>	#23877 ✓ refs/merge-r... → 752b3333	#9270 by <div></div>	lint	eslint	<div>00:00:36</div> <div>10 minutes ago</div>	
<div>passed</div>	#23876 ✓ refs/merge-r... → 752b3333	#9270 by <div></div>	build	install_dependencies	<div>00:00:42</div> <div>11 minutes ago</div>	<div></div>
<div>passed</div>	#23824 ✓ refs/merge-r... → 14d7b998	#9244 by <div></div>	test	testing	<div>00:00:29</div> <div>2 hours ago</div>	<div></div>
<div>passed</div>	#23823 ✓ refs/merge-r... → 14d7b998	#9244 by <div></div>	lint	eslint	<div>00:00:36</div> <div>2 hours ago</div>	<div></div>
<div>passed</div>	#23822 ✓ refs/merge-r... → 14d7b998	#9244 by <div></div>	build	install_dependencies	<div>00:00:43</div> <div>2 hours ago</div>	<div></div> <div></div>
<div>passed</div>	#23814 ✓ refs/merge-r... → 47bba5f6	#9239 by <div></div>	test	testing	<div>00:00:34</div> <div>2 hours ago</div>	<div></div>
<div>passed</div>	#23813 ✓ refs/merge-r... → 47bba5f6	#9239 by <div></div>	lint	eslint	<div>00:00:36</div> <div>2 hours ago</div>	<div></div>
<div>passed</div>	#23812 ✓ refs/merge-r... → 47bba5f6	#9239 by <div></div>	build	install_dependencies	<div>00:00:43</div> <div>2 hours ago</div>	<div></div> <div></div>
<div>passed</div>	#23810 ✓ refs/merge-r... → 47bba5f6	#9239 by <div></div>	test	testing	<div>00:00:24</div> <div>2 hours ago</div>	

(Notice the bottom job is the failed testing job for pipeline #9236)

- Pipeline evidence (for a specific pipeline):

<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/pipelines/9271>

3760f20 > chatron > wiki-chatbot > Pipelines > #9271

passed Pipeline #9271 triggered 5 minutes ago by Seegal Panchal

Merge branch 'seegal' into 'master'

Refactoring of command parsing and new tests

See merge request !9

3 jobs for master in 1 minute and 47 seconds (queued for 3 seconds)

latest

d9da41bb

No related merge requests found.

Pipeline DAG Beta Jobs 3

Build Lint Test

install_dependen... eslint testing

- Also, please take note of individual merge requests also receiving pipelines for each commit besides when the merge request branch is merged with master (4 commits, 5 pipelines):

https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/8/pipelines

3760f20 > chatron > wiki-chatbot > Merge Requests > 18

Merged Opened 7 hours ago by Seegal Panchal Edit Report abuse

Added /playerLowest functionality

Overview 0 Commits 4 Pipelines 5 Changes 3

Status	Pipeline	Triggerer	Commit	Stages	
	#9172		d2c225f0 Merge branch 'seegal' wit...		00:02:08 7 hours ago
	#9171 latest		060e69a5 Merge branch 'seegal' in...		00:01:52 7 hours ago
	#9170		ea414962 Merge branch 'seegal' in...		00:01:50 7 hours ago
	#9168		123db96d Merge branch 'seegal' in...		00:01:52 7 hours ago
	#9167		9576231a Merge branch 'seegal' in...		00:01:30 7 hours ago

(Notice that pipeline #9167 failed ESLint lint stage)

- For example, if we take a look at the most recent pipeline where merge request 9 was merged to master, we can see the following for the 3 jobs for pipeline #9271:
 - Build: Install Dependencies -
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23879>
 - Lint: ESLint -
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23880>
 - Test: Mocha Testing + Istanbul Coverage Report -
<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/jobs/23881>
- For the above links, all are success cases. To see some more related evidence, please see the related screenshots in the following sections.

- **ESLint - Linter**

- Linting is powerful because it allows us to standardize the formatting of our code, making it easy to read. By using ESLint, we can easily run a linter on all of our JS files, in addition to being able to add it to the GitLab CI/CD pipeline as a linter job, which automates the linting of the code. Moreover, we can simply use the “--fix” command on ESLint to fix simple formatting issues, so that the developer does not need to refactor the code manually. Below see a success case compared to a failed case for the linting ESLint job from the pipeline:

```
$ eslint --ignore-path .lintignore .  
Saving cache  
Creating cache default-3...  
node_modules/: found 19505 matching files  
No URL provided, cache will be not uploaded to shared cache server. Cache will be stored only locally.  
Created cache  
Job succeeded
```

```
$ eslint --ignore-path .lintignore .  
/builds/3760f20/chatron/wiki-chatbot/bot.js  
  81:21  error  Unexpected lexical declaration in case block  no-case-declarations  
✖ 1 problem (1 error, 0 warnings)  
ERROR: Job failed: exit code 1
```

(NOTE: also see image in GitLab CI/CD Pipeline that shows the job of the second stage/phase of the pipeline is for linting using ESLint. Here it is referenced in our CI file:

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/master/.gitlab-ci.yml#L23>)

- **Mocha + Istanbul - Testing + Coverage Reports**

- Mocha is an industry standard unit testing tool, which allows us to write an easily understandable “English” testing suite. Combined with NYC (Istanbul), the unit tests can track which parts of the codebase are executed, allowing us to know exactly how much of our code is covered by the tests (that are run using Mocha).

- Testing: (example of some tests succeeding and failing for the Command part)

<pre> Command /player LeBron James ✓ should return a string (136ms) /playerSingleStat STL James Harden ✓ should return a string ✓ should return 125 STL /playerSingleStat with bad input ✓ should reject the input /playerHighest STL ✓ should return a string ✓ should return James Harden /playerHighest VOL ✓ should reject the input /playerLowest STL ✓ should return a string ✓ should return Udonis Haslem /playerLowest VOL ✓ should reject the input /help ✓ should return the help string /helpStats ✓ should return the help string attempt to analyze a non-command ✓ should reject the input </pre>	<pre> Command /player LeBron James ✓ should return a string (133ms) /playerSingleStat STL James Harden ✓ should return a string ✓ should return 125 STL /playerSingleStat with bad input ✓ should reject the input /playerHighest STL ✓ should return a string 1) should return James Harden /playerHighest STL ✓ should reject the input /playerLowest STL ✓ should return a string ✓ should return Udonis Haslem /playerLowest STL ✓ should reject the input /help ✓ should return the help string /helpStats ✓ should return the help string attempt to analyze a non-command ✓ should reject the input </pre>
--	---

(Notice the second check in the fourth test failed - this is later described in more detail before the coverage report, as seen below:)

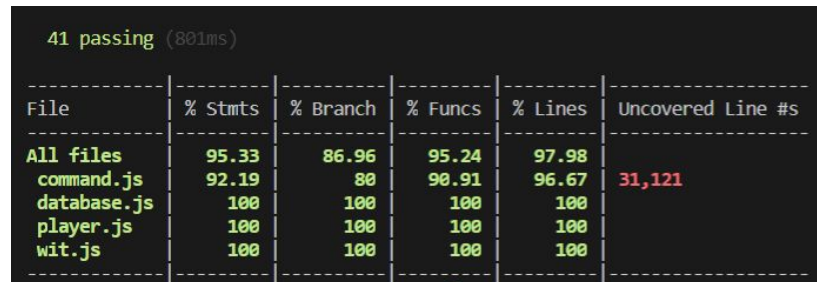
```

40 passing (1s)
1 failing
1) Command
   /playerHighest STL
     should return James Harden:
     AssertionError [ERR_ASSERTION]: The expression evaluated to a falsy value:
     assert(response === 'Player \'James Harden\' has the highest STL at 125')
       + expected - actual
       -false
       +true

       at Context.it (test/command.test.js:39:13)
       at process._tickCallback (internal/process/next_tick.js:68:7)
-----|-----|-----|-----|-----|-----
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files | 95.33   | 86.96    | 95.24    | 97.98    |
command.js | 92.19   | 80       | 90.91    | 96.67    | 31,121
database.js | 100     | 100      | 100      | 100      |
player.js  | 100     | 100      | 100      | 100      |
wit.js     | 100     | 100      | 100      | 100      |
-----|-----|-----|-----|-----|-----
npm ERR! Test failed.  See above for more details.
ERROR: Job failed: exit code 1

```

- Coverage Report: (success case, colour present because not screenshot from the pipeline job although it's the same data)



```
41 passing (801ms)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	95.33	86.96	95.24	97.98	
command.js	92.19	80	90.91	96.67	31,121
database.js	100	100	100	100	
player.js	100	100	100	100	
wit.js	100	100	100	100	

- NOTE: also see image in GitLab CI/CD Pipeline that shows the job of the third stage/phase is for running tests (using Mocha) which also generates a coverage report after (using Istanbul). Here it is referenced in our CI file:

<https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/blob/master/.gitlab-ci.yml#L32>

2. DevOps

NOTE: see Team Standards doc for a brief merge and testing discussion in *#committing-code*: <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/TeamStandards>

2.1 CI/CD - Employed for Linter and Unit Tests

GitLab pipelines are employed for ANY commit to a merge request. Additionally, a GitLab merge pipeline is executed on an attempted merge, and will only merge if the pipeline passes successfully. This includes testing the linter, ESLint, as the second job, succeeding the build job, and proceeding the testing job (which also generates the coverage report).

- Please see section “*1.4 Testing Tools and Justification of Choice*” above for lots of evidence and screenshots of the CI/CD being employed, including notes about the individual jobs conducted during the pipeline for both successes and failures.

2.2 Merge Rules

Please refer to our Team Standards document (<https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/wikis/TeamStandards>), for our branching and merging strategy, as well as a more detailed description about merge rules for our team. Although, we do highlight some specific things below.

Also, please note that our team’s merge rules specify that pipeline success is required in order to merge. If the pipeline fails for the merge to master, a merge request fails and is aborted so the developer(s) can fix issues and retry. Also note that the pipeline runs on each branch whenever code is pushed there.

In regards to including a link and/or sample of merge request template used by the team, please see our GitLab general settings, where “Merge requests” section is expanded: <https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/edit>. Scrolling to the bottom of this

section, you will see our description template for merge requests:

Default description template for merge requests [?](#)

Brief title of merge request

- try to tag issue number with #NUM
- remember to tag any related epics (i.e. add that label)
- remember to tag related milestone

Description parsed with [GitLab Flavored Markdown](#)

Merge request approvals

Set a number of approvals required, the approvers and other approval settings. [Learn more about approvals.](#)

In addition, take note that the description of each of our merged merge requests for milestone 2 adhered to this template. For example, merge request 9:

https://git.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/9, which had a description template of the following:

3760f20 > chatron > wiki-chatbot > Merge Requests > 19

Merged Opened 3 hours ago by  **Seegal Panchal**

Refactoring of command parsing and new tests

Overview 4 **Commits** 4 **Pipelines** 5 **Changes** 8

This is a massive refactor adding a new class called Command which handles command (!/) inputs

- Issue #6, #12, #13, #35
- Epic: Bot Inquiries
- Milestone 2

2.3 Merge Discussion and Resolution Examples

- Merge Request #2
 - https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/2
 - Jason comments on the merge request suggesting additional comments and more descriptive variable names, in order to adhere to our coding standards. Seegal resolves these issues. The merge request is then approved, and the changes were merged into the master branch. Evidence below.



Jason Hung @jhung04 started a thread on an old version of the diff 3 weeks ago
Resolved by Seegal Panchal 1 day ago

Toggle thread

bot.js

```
1 + // Copyright (c) Microsoft Corporation. All rights reserved.  
2 + // Licensed under the MIT License.  
3 +  
4 + const { ActivityHandler, MessageFactory } = require('botbuilder');  
5 + const NBA = require("nba");  
6 +  
7 + class EchoBot extends ActivityHandler {  
8 +   constructor() {  
9 +     super();  
10 +    // See https://aka.ms/about-bot-activity-message to learn more about the message and other activity types.  
11 +    this.onMessage(async (context, next) => {  
12 +      //const replyText = `Echo: ${ context.activity.text }`;  
13 +      //let stat = nba.getPlayerID(context.activity.text);  
14 +      var res = context.activity.text.split(" ");  
15 +  
16 +      if (res[0] === "/player") {
```



Jason Hung @jhung04 · 3 weeks ago

Maintainer

Please add a comment on what this block of code does or make a function where the name describes its function.



Jason Hung @jhung04 · 3 weeks ago

Maintainer

Also give res a bit more descriptive name



Seegal Panchal @panchals changed this line in version 3 of the diff 1 week ago

Merge request approved. Approved by

[View eligible approvers](#)

Merged by Seegal Panchal 1 day ago

Revert

Cherry-pick

The changes were merged into master with fe3e68d0

The source branch has been deleted

Mentions #18 and #6

- Merge Request #3

- https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/3
- Jason comments on the merge request suggesting the implementation of an error message in the event that a command is not recognized. Seegal resolves this issue by implementing an error message for the specified case. Evidence below.



Jason Hung @jhung04 started a thread on an old version of the diff 23 hours ago
Resolved by Seegal Panchal 23 hours ago

[Toggle thread](#)

bot.js

```
15 -     const player = new Player();
16 -     let response = await player.getPlayer(context.activity.text);
17 -     if (typeof response === 'string') {
18 -         await context.sendActivity(MessageFactory.text(response, response));
19 -     } else {
20 -         for (var key in response[0]) {
21 -             let msg = key + ": " + (response[0])[key];
22 -             await context.sendActivity(MessageFactory.text(msg, msg));
23 -         }
24 +
25 +     const player = new Player();
26 +     let res = context.activity.text.split(" ");
27 +     let command = res[0];
28 +     res.shift();
29 +     let playerName = res.join(" ");
30 +
31 +     switch(command) {
```



Jason Hung @jhung04 · 23 hours ago

Maintainer

Is there a generic error message if a command isn't recognized or does not exist?



Seegal Panchal @panchals · 23 hours ago

Maintainer

I didn't think of that, I have added one now. It will respond with "Command not found."



Seegal Panchal @panchals changed this line in [version 2 of the diff](#) 23 hours ago

- Jason comments on the merge request suggesting the addition of comments describing a block of code, in order to adhere to our coding standards. Seegal resolves this by adding the suggested comments into the code. The merge request is then approved, and the changes were merged into the master branch. Evidence below.



Jason Hung @jhung04 started a thread on an old version of the diff 23 hours ago
Resolved by Seegal Panchal 23 hours ago

[Toggle thread](#)

```
bot.js
16 - let response = await player.getPlayer(context.activity.text);
17 - if (typeof response === 'string') {
18 -   await context.sendActivity(MessageFactory.text(response, response));
19 - } else {
20 -   for (var key in response[0]) {
21 -     let msg = key + ": " + (response[0])[key];
22 -     await context.sendActivity(MessageFactory.text(msg, msg));
23 -   }
24 +
25 + const player = new Player();
26 + let res = context.activity.text.split(" ");
27 + let command = res[0];
28 + res.shift();
29 + let playerName = res.join(" ");
30 +
31 + switch(command) {
32 +   case '/player':
```



Jason Hung @jhung04 · 23 hours ago

Maintainer



Consider adding a small comment here describing what it does (e.g. get all player stats)



Seegal Panchal @panchals · 23 hours ago

Maintainer



Done



Seegal Panchal @panchals changed this line in [version 2](#) of the diff 23 hours ago



Merge request approved. Approved by

[View eligible approvers](#)



Merged by Seegal Panchal 23 hours ago [Revert](#) [Cherry-pick](#)

The changes were merged into `master` with `c060e226`

The source branch has been deleted

Mentions [#12](#), [#18](#), and [#6](#)

- Merge Request #6


- https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/6
- Mitchell comments on the merge request suggesting descriptive comments on the async functions in the code, in order to adhere to our coding standards. Seegal resolves this by adding the suggested comments to the code. Evidence below.



 **Mitchell Van Braeckel** @mvanbrae · 20 hours ago
Resolved by **Mitchell Van Braeckel** 20 hours ago
on async functions, add a brief description comment

▼ Collapse replies


 **Seegal Panchal** @panchals · 20 hours ago
will add function comments now

- Mitchell comments on the merge request suggesting a new parameter to be added to code querying the database - he mentions that in the future, a parameter specifying “Season” will be necessary. Seegal resolves this by adding the suggested table parameter. The merge request is then approved, and the changes were merged into the master branch. Evidence below.

 **Mitchell Van Braeckel** @mvanbrae started a thread on an old version of the diff 20 hours ago
Resolved by **Mitchell Van Braeckel** 20 hours ago ⤴ Toggle thread

 **impl/player.js** 

14	14	});
15	15	});
16	16	}
17	+	async getPlayerSingleStatCommand(playerName, stat) {
18	+	

 **Mitchell Van Braeckel** @mvanbrae · 20 hours ago
needs table param, use for db.findPlayerSingleStat line 21 - eventually will have multiple seasons

 **Seegal Panchal** @panchals · 20 hours ago
will add table parameter

 **Seegal Panchal** @panchals changed this line in [version 2 of the diff](#) 20 hours ago

 **Merge request approved.** Approved by 

► [View eligible approvers](#)

✓ Merged by  **Seegal Panchal** 20 hours ago [Revert](#) [Cherry-pick](#)

The changes were merged into `master` with `b0dec01c` 


The source branch has been deleted

Mentions [#12](#), [#18](#), [#36](#), and [#37](#)

- Merge Request #7

- https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/7

- Jason comments on the merge request suggesting more descriptive comments for each test, in order to adhere to our coding standards. Seegal resolves this by adding the suggested comments identifying the tests. The merge request is then approved, and the changes were merged into the master branch. Evidence below.


 Jason Hung @jhung04 started a thread on the diff 5 hours ago
Resolved by Seegal Panchal 4 hours ago Toggle thread

test/database.test.js


```

5   5   describe('Database', function() {
6   6   describe('Find player Kim Kardashian', function() {





```

 Jason Hung @jhung04 · 5 hours ago Maintainer ✓ 😊 ✎ ⋮


Consider adding a comment above each test, labelling it with a number so it is easier to reference in the milestone documents (e.g. "// Test 1") rather than by line number which can change. This comment also applies for other test files. Test numbers don't have to be unique across all test files, just unique within a single test file. (For example we could reference it as "database.test.js Test 2")


 Seegal Panchal @panchals · 4 hours ago Maintainer ✓ 😊 ✎ ⋮

Added comments identifying the tests

 Merge request approved. Approved by   

[View eligible approvers](#)

✓ Merged by  Seegal Panchal 5 hours ago Revert Cherry-pick

The changes were merged into master with [2e852962](#) 

Mentions #12, #13, #18, and #6

● Merge Request #9

- https://gitlab.socs.uoguelph.ca/3760f20/chatron/wiki-chatbot/-/merge_requests/9
- Mitchell comments on the merge request suggesting an improvement to the code, involving the removal of single quote characters surrounding a player's name when the answer is returned. Seegal resolves this by removing the single quotes in the specified code. Evidence below.



Mitchell Van Braeckel @mvanbrae started a thread on an old version of the diff 12 minutes ago
Resolved by Seegal Panchal 10 minutes ago

Toggle thread

impl/command.js

```
129 +         .then((response) => {  
130 +             if (typeof response === 'string') return resolve(response);  
131 +             else return resolve(`Player ${ playerName } has ${ response[0][stat] } ${ stat.toUpperCase() }`);  
132 +         })  
133 +         .catch((error) => reject(error));  
134 +     });  
135 + }  
136 +  
137 + async getHighestLowestCommand(stat, order) {  
138 +     return new Promise((resolve, reject) => {  
139 +         const player = new Player();  
140 +         player.getStatHighestLowestPlayer('s19', stat, order)  
141 +             .then((response) => {  
142 +                 if (typeof response === 'string') return resolve(response);  
143 +                 else {  
144 +                     if (order === 'DESC') resolve(`Player '${ response[0].PLAYER_NAME }' has the highest ${ stat`  
    } at ${ (response[0])[stat] }`);
```



Mitchell Van Braeckel @mvanbrae · 12 minutes ago

Maintainer ✓ 😊 ✎ ⋮

We don't need to surround player name in single quotes when we return the answer



Seegal Panchal @panchals · 10 minutes ago

Maintainer ✓ 😊 ✎ ⋮

will be fixed in the next commit



Seegal Panchal @panchals changed this line in [version 2 of the diff](#) 7 minutes ago

- Mitchell comments on the merge request pointing out a typo in the code - “STL” was written when it should be “VOL”. Seegal resolves this by fixing the typo. The merge request is then approved, and the changes were merged into the master branch. Evidence below.



Mitchell Van Braeckel @mvanbrae started a thread on an old version of the diff 1 minute ago
Resolved by Seegal Panchal 48 seconds ago

[Toggle thread](#)

test/command.test.js

```
28 +     assert.rejects(response);
29 +   });
30 + });
31 + // Test 3
32 + describe('/playerHighest STL', function() {
33 +   it('should return a string', async () => {
34 +     const response = await command.analyze('/playerHighest STL');
35 +     assert(typeof response === 'string');
36 +   });
37 +   it('should return James Harden', async () => {
38 +     const response = await command.analyze('/playerHighest STL');
39 +     assert(response === 'Player James Harden has the highest STL at 125');
40 +   });
41 + });
42 + // Test 3 - bad input
43 + describe('/playerHighest STL', function() {
```



Mitchell Van Braeckel @mvanbrae · 1 minute ago

Maintainer

as well as test 4 - bad input: the describe() line should also use VOL instead of STL



Seegal Panchal @panchals · 52 seconds ago

Maintainer

will be fixed in the next commit



Seegal Panchal @panchals changed this line in [version 4 of the diff](#) 12 seconds ago

Merge request approved. Approved by

[View eligible approvers](#)

Merged by Seegal Panchal 12 hours ago [Revert](#) [Cherry-pick](#)

The changes were merged into [master](#) with [d9da41bb](#)

The source branch has been deleted

Mentions [#12](#), [#13](#), [#35](#), and [#6](#)