BoilerBot

Team 1: Aditya Dhingra, Rahul Pai, Eehita Parameswaran,
Kush Rustagi, Devaunsh Sambhav, Yash Shiroya

March 6th, 2017

Sprint 2 Planning

Contents

1 Sprint Overview

1.1 Overview

This sprint works on the direct extension of the previous bot setup and plans to focus on updating the user with Purdue Crime Statistics. The critical step in the sprint will be to scrape the Purdue Crime Statistics page, then allocate storage(another model) for the statistics and respond to the user with the last collected statistics which will be updated periodically. The crime statistics include reports such as daily crime statistics, active warrants and last committed crimes. The crime statistics features will contribute a much required safety aspect to the service which work well with the convenience features of the previous and upcoming sprints.

1.2 Scrum Master

Eehita Parameswaran

1.3 Meeting Schedule

Tuesday, Thursday, Saturday at 6:00PM

1.4 Risks or Challenges

One of the main challenges in this sprint is to integrate the individual components and ensure proper functionality thereafter. A large amount of time in this sprint will be spent on building a faster implementation of our Natural Language Processing algorithm. Training the bot to better understand what the user is trying to say is also a challenge as we need to make various intents. Creation of a web scraper and testing the web scraper will be the first risk since none of us have done it before. The next problem will be with separating the data and storing it in the database. It may be difficult using web scraping to properly separate the crime statistics by latest, monthly, yearly, and active warrants. Then saving each part into the database provides another challenge in properly accessing each one. The next set of risks revolves around server optimization, right now it takes a while for the bot to activate and begin responding. Also we need to make sure that we have a standard message to send to users when the server is down for maintenance.

1.5 Testing Framework

Team Boilerbot tried to automate most of the tests built in Node.js. We used a tool called 'Mocha' which is a feature-rich JavaScript test framework running on Node.js and in the browser. This tool helped us make asynchronous tests in a simple manner. These Mocha tests run serially thus, allowing for flexible and accurate reporting, while mapping uncaught

exceptions to the correct test cases. So, with Mocha we had an environment to make our tests but to actually test our HTTP calls, we required an add-on library.

Hence, to add the necessary logic, we utilized 'Chai' which is an assertion library. One of the main reasons we chose Chai over other assertion libraries was because it allowed us to choose the type of assertion style we'd like to use. This library comes with three different assertion types. It has the *should* style, the *expect* style and the *assert* style. Mocha essentially provides *traction* for unit testing as there are a lot of libraries that are built on the *expect* package such as supertest. Supertest allows developers to test API endpoints very easily by querying the api directly and asserting the responses. The *expect* package can then be used to test the responses in more detail. We decided to use the *expect* style and by using chai-http, we were able to make the actual HTTP requests. And then, we tested the responses with the expected results.

2 Current Sprint Detail

2.1 User Story #9

As a user, I would like to adjust the settings across the conversation.

Task	Time(hrs)	Assigned To
Validate that the bot is keeping a log of all the conversations held.	6	Devaunsh
Add stored procedures to ensure that conversations are being mapped to the correct user.	8	Rahul
Test stored procedures using automated testing framework (mocha-chai) and ensure that correct response is returned.	2	Eehita

2.1.1 Acceptance Criteria

- Given that the user is able to converse with the bot, when the user wants to see the history of the conversation, then the user should be able to see the log of all the conversations held till date.
- Using Mocha-Chai testing framework, we intend to test the stored procedures to ensure that each conversation is being mapped to the appropriate user by checking that it's a one-to-one mapping .

2.2 User Story #10

As a user, I expect the bot to keep a log of all the messages across the conversation.

Task	Time(hrs)	Assigned To
Check that the bot is keeping a log of all the settings made by the user.	6	Devaunsh
Add stored procedures to ensure that settings are being mapped to the correct user.	8	Rahul

2.2.1 Acceptance Criteria

- Given that the user is able to converse with the bot, when the user makes certain settings to the default settings in the conversational state of the bot, then those settings should not get eradicated until the user changes it.
- Using Mocha-Chai testing framework, we intend to test the stored procedures to ensure that each user setting is being mapped to the appropriate user by checking that it's a one-to-one mapping.

2.3 User Story #11

As a first-time user, I would like to have a tutorial on how the bot works.

Task	Time(hrs)	Assigned To
Make recast.ai understand when the user wants help.	6	Yash
Create messages in a tutorial function.	6	Eehita
Display tutorial message when the user's intent is help.	6	Eehita

2.3.1 Acceptance Criteria

- Given that the the Recast.ai is setup, when the user asks for help, then the Recast.ai should be able to understand that the user is asking for help.
- Given that the bot is able to communicate with the user, we should create a set of messages that gives the user detailed information about the bot's functionality and usage.
- Given that the tutorial function is set up, when the user asks for help, then the chatbot should call the tutorial function and we will test this by rendering the set of messages that give information about the usage and functionality of the chatbot.
- Test feature manually by sending an input of 'help' or and checking the output against the predefined output. Another test case would be when a new user tries to use Boilerbot, a tutorial should show up for the user.

2.4 User Story #12

As a user, I would like the bot to reply quickly.

Task	Time(hrs)	Assigned To
Make the bot send intermediary messages such the developer team is informed about the bot's response time.	4	Kush
Send a typing message/emoticon while we are waiting for BoilerBot to respond.	4	Kush

2.4.1 Acceptance Criteria

- Given that the user has been waiting for a reply for a while, we would send an intermediary message i.e. typing dots to ensure that the user knows that the bot will reply soon.
- Test feature by sending multiple messages to BoilerBot and ensuring that our bot is 'typing' such that the user knows that a response will be received soon.

2.5 User Story #13

As a user, I expect the bot to inform me about server downtime.

Task	Time(hrs)	Assigned To
Create a downtime message in a separate function.	7	Aditya
Get all user id's from database.	8	Devaunsh
Send downtime message to them using downtime functions.	7	Aditya

2.5.1 Acceptance Criteria

- Given that the chatbot is functional, we have to create a function that would render a message to a user regarding when the server downtime is expected.
- Given the database is functional, when we are planning to stop the server, a function should get all the user id's present so that we can notify all users of our chatbot about the server downtime.
- Given that the function to notify about server downtime is implemented, when we are planning to stop the server, we'll test the response in Boilerbot and ensure that the function sends the message about downtime to all the users listed in our database.
- Test using automated testing framework such that when the server is shut down by the developers for sometime, the tester automatically gets a notification about the downtime message.

2.6 User Story #14

As a user, I expect the bot to be versatile.

Task	Time(hrs)	Assigned To
Validate that the bot works has appropriate responses on all platforms and to almost all queries.	4	Rahul

2.6.1 Acceptance Criteria

- Given that the chatbot is functional, we will using Natural Language Processing to ensure that Boilerbot is trained well to answer almost any query and it responds appropriately to unusual queries.
- Test feature using mocha-chai framework and send varied messages(help, feedback etc.) to the bot to ensure that Boilerbot responds with an appropriate message everytime (tested against valid output)

2.7 User Story #15

As a user, I would like to get information regarding the last crime committed at Purdue.

Task	Time(hrs)	Assigned To
Setup connection and integrate Purdue Crime stats page.	6	Yash
Write a web scraper to scrape crime statistics from the Purdue University Police Department web page.	6	Yash
Separate the latest crime statistic from the scraped data.	8	Kush
Display the crime statistic using a generic messenger template (eg: cards).	8	Kush

2.7.1 Acceptance Criteria

- Given that Recast.ai is set up, when a user sends a message, then it would be sent to the API for parsing.
- Given that we have scraped all of the data from the Purdue University Police Department page, we will be able to keep all of the saved crime statistics in the database.
- Given that the user requests the latest crime statistic, access the latest crime statistic from the database.
- Test feature by using Mocha-Chai testing framework and check whether or not we get information about the last crime committed at Purdue when the user sends request like 'last crime?'.

2.8 User Story #16

As a user, I would like to get monthly crime statistics for the current year at Purdue.

Task	Time(hrs)	Assigned To
Setup connection and integrate Purdue Crime stats page.	8	Eehita
Scrape crime statistics from Purdue University Police Department page.	8	Eehita
Separate crime statistics for the current month from overall statistics.	8	Aditya
Display the monthly crime statistics using a generic messenger template (eg: lists).	8	Aditya

2.8.1 Acceptance Criteria

- Given that Recast.ai is set up, when a user sends a message, then it would be sent to the API for parsing.
- Given that we have scraped all of the data from the Purdue University Police Department page, we will be able to keep all of the saved crime statistics in the database.
- Given that the user requests monthly crime statistics, accesses the monthly crime statistics from the database.
- Test feature by using Mocha-Chai testing framework and check whether or not we get the monthly crime statistics for the current year at Purdue when the user sends request like 'monthly crime stats'.

2.9 User Story #17

As a user, I would like to know about any active warrants at Purdue.

Task	Time(hrs)	Assigned To
Setup connection and integrate Purdue Crime stats page.	6	Devaunsh
Scrape active warrants data from the Purdue University Police Department web page.	6	Devaunsh
Display the active warrants using a generic messenger template (eg: cards).	4	Rahul

2.9.1 Acceptance Criteria

- Given that Recast.ai is set up, when a user sends a message, then it would be sent to the API for parsing.
- Given that we have scraped all of the data from the Purdue University Police Department page, we will be able to keep all of the saved crime statistics in the database.
- Given that the user requests active warrants, accesses the active warrants saved in the database.
- Test feature by using Mocha-Chai testing framework and check whether or not we get correct information about the active warrants committed at Purdue when the user sends request like 'active warrants?' .

2.10 User Story #18

As a user, I would like to get yearly crime statistics for the current year at Purdue.

Task	Time(hrs)	Assigned To
Setup connection and integrate Purdue Crime stats page.	4	Yash
Scrape crime statistics from Purdue University Police Department web page.	4	Yash
Separate crime statistics for the current year from overall statistics.	6	Kush
Display the yearly crime statistics using a generic messenger template (eg: lists).	6	Kush

2.10.1 Acceptance Criteria

- Given that Recast.ai is set up, when a user sends a message, then it would be sent to the API for parsing.
- Given that we have scraped all of the data from the Purdue University Police Department page, we will be able to keep all of the saved crime statistics in the database.
- Given that the user requests active warrants, accesses the active warrants saved in the database.
- Test feature by using Mocha-Chai testing framework and check whether or not we get correct information about the active warrants committed at Purdue when the user sends request like 'active warrants?' .

2.11 Developer Story #1

As a developer, I would like to come up with regular updates to the chatbot. Updates that include giving more relevant information based on user feedback.

Task	Time(hrs)	Assigned To
Analyse the feedback given by the users.	8	Eehita
Train Recast.ai to understand the suggestion intent.	8	Eehita
Inform the users about the updates as soon as they roll out.	10	Aditya

2.11.1 Acceptance Criteria

- Given that the user sent their recommendations, then use the stored user recommendations to analyze user likes and dislikes as a whole.
- Given that Recast.ai is trained to understand the suggestion intent, the bot can send the appropriate message to the user accordingly.
- Given that the bot is updated to a particular version, then all users receive updates immediately.
- Test analytics manually to ensure that our algorithm is saving the feedback correctly to the database and we are able to come up with correct assumptions about user suggestions.

2.12 Developer Story #2

As a developer, I would like my bot to handle at least 100 users at once.

Task	Time(hrs)	Assigned To
Learn about multi-threading in Node.js.	8	Rahul
Implement functionality to serve the tasks asynchronously.	8	Rahul
Check for regular server upgrades.	4	Kush

2.12.1 Acceptance Criteria

- Given that multiple users are using the bot at the same time, the bot should be able to serve their individual requests concurrently and in a timely manner.
- Given that multiple users send multiple queries/texts i.e. more than can just handling it concurrently, the bot should be able to efficiently queue tasks so as not to cause clashes between multiple users and their respective data.
- Given that the server is running a specific version, when the new version of the server comes out, then check the version and any issues pertaining to it and upgrade to it accordingly.
- Test feature by using automated testing framework of mocha-chai such that there are at least 100 'pseudo-users' who access/send Boilerbot a message/query at the same time.

2.13 Developer Story #3

As a developer, I would reduce the downtime for maintenance to 2 hours per month.

Task	Time(hrs)	Assigned To
Implement and test the functionality.	8	Devaunsh
Send users a message about the server downtime.	8	Yash

2.13.1 Acceptance Criteria

- Given that the user base increases, then we would like to have a fast maintenance pattern so as not to hinder user experience for very long.
- Given that the user is providing the feedback, then analyze the feedback and implement the functionality in a 2 hour time span once a
- Given that the some new functionality is implemented and is in the testing phase, when the update is due, then inform the all the users about the time during which the application will not be available.
- Test feature by setting the downtime for maintenance to around 10 mins and check whether or not it takes place by receiving a console log message. Developers can then take necessary action accordingly.

3 Remaining Backlog

3.0.1 Functional Requirements

USER:

1. As a user, I would like bot to provide me with information about the opening and closing times of dining courts at Purdue.
2. As a user, I would like the bot to provide me with information about food at dining courts at Purdue.
3. As a user, I want the bot to have information about the courses being offered in the current semester.
4. As a student at Purdue, I would like to know the seats left in a course that I want to register for.
5. As a student at Purdue, I would like the bot to provide me with information about the different professors teaching a course.
6. As a student at Purdue, I would like the bot to provide me with information about the different rooms that a specific course will be taught in.
7. As a student at Purdue, I would like the bot to provide me with information about the different times at which a specific course is being offered.
8. As a student at Purdue, I would like the bot to give me details like the credits offered for taking a course.
9. As a user, I want to be able to submit features that I would like to see on the bot.

DEVELOPER:

1. As a developer, I would like to be able to handle the UI for mobile and web appropriately.
2. As a developer, I would like to implement further fixes and enhancements according to the usage of the user.
3. As a developer, I would like to be able to implement popular feature requests submitted by students.

3.0.2 Non-Functional Requirements

1. As a user of the chatbot, I expect my information to be secure from other users.
2. As a developer, I expect the chatbot to be scalable. The performance shouldn't be affected with increasing user base.
3. As a developer, I would like the Natural Language Processing model to be reusable and extensible.
4. As a developer, I would like to securely store user usage and telemetry for further understanding of user requirements without storing user credentials.
5. Build a database that is computationally cheaper to run and maintain.