BoilerBot

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

February 13th, 2017

Sprint 1 Planning

Contents

1 Sprint Overview

1.1 Overview

In this sprint we will largely focus on building a minimum viable product of our chat bot. Issues like performance and scalability will be considered in the next sprint. For now, we would like to have all components functioning correctly and working well together, so that we can build upon them in the next sprint. The core user stories of our chatbot will be dealt with first, followed by its supporting user stories. The user should be able to converse with the bot and ask the bot for dining court locations, especially which court is closest to the user at that particular moment. Additionally, we hope to start working on the algorithm that would scrape the crime statistics page, along with the API processing that is to be used for the next two sprints; this implementation would be improved in every sprint.

1.2 Scrum Master

Eehita Parameswaran

1.3 Meeting Schedule

Tuesday, Thursday, Saturday at 6:00PM

1.4 Risks or Challenges

Most of our team members are not familiar with all the technologies. So, a lot of our time in the first sprint will be spent on getting familiar with the technologies and web frameworks like Python, Flask, REST APIs and MySQL. It is also essential that our core component is working properly such that our team members can work individually, rather remotely, for the future sprints. Lastly, this is the first time we are working together on a project, so it would definitely take a while to understand each team member's work ethic and we intend to start working like a well-oiled machine soon.

2 Current Sprint Detail

2.1 User Story #1

As a first-time user, I would like the bot to introduce itself and provide me a set of instructions.

Task	Time(hrs)	Assigned To
Stored procedures to query the history of a user from the database.	4	Devaunsh
Display information about the functionality of bot to the new user.	8	Rahul
Update entry in database with new user information.	4	Devaunsh
Take advantage of Recast.ai to give user a personalized response.	4	Eehita
Make Recast.ai understand the user's message and reply appropriately.	10	Eehita

2.1.1 Acceptance Criteria

- Given that our database is up and running, when a message is received from a user, then we query the database to check whether the user exists or is a new user.
- Given that the query returns history of the user, if the user is new, then the chatbot should introduce itself and state its functionality.
- Given that our database is setup, when new user connects with our chatbot, then our database will store the new user's information.
- Given that our Recast.ai i.e. Natural Language Processing API is set up, when a message is received from a user, then our Recast.ai should understand the user's message and an appropriate text is returned to the user.

2.2 User Story #2

As a user, I would like the bot to exchange pleasantries with me.

Task	**Time(hrs)**	**Assigned To**
Use Recast.ai to create a list of user intents.	8	Kush
Recognize preliminary user intent with Recast.ai.	4	Yash
Create training examples of specific greetings.	6	Yash
Caching user's intent and entity.	8	Yash
Implement functionality to respond with pleasant greeting.	4	Yash

2.2.1 Acceptance Criteria

- Given that Recast.ai is setup, once a user sends a message it will be sent to the API for parsing and extraction of intent.
- Given that a list of intents has been set up, the API will be able to understand and group the message by intent.

Ex. What **food** is being served at **Ford**?
Recast -> (intent: **food**), (entities: **Ford**)

- Given that the API understands the message, the user's message would be parsed and the intent with possible entities would be sent to the user.
- Given that the API can send replies, the parsed reply would be received by the backend and used to send the user the correct response.

2.3 User Story #3

As a user, I would like to interact with the bot.

Task	Time(hrs)	Assigned To
Initialize Project in Flask.	6	Aditya
Add directory structure.	2	Aditya
Setup Heroku account and sharing privileges.	4	Kush
Push code to Heroku.	4	Kush
Setup SQL adapter for Python(SQLAlchemy).	6	Devaunsh
Design database schema.	6	Aditya
Adapt MySQL schema.	4	Eehita
Implement SQLAlchemy scripts.	8	Aditya
Deploy MySQL scripts.	6	Rahul
Setup Recast.ai for Natural Language Processing.	6	Eehita
Bot performs 'echo' response and dynamic Heroku dyno reactivation.	8	Yash

2.3.1 Acceptance Criteria

- Given that the backend is created, when a new user signs into our bot, then a new database entry will be created that includes details like their Facebook user ID, name, etc.

- Given that MySQL and the Flask framework is setup, when the user sends a message to the bot, then the user information would be stored in the database.

- Given that Heroku is set up and running with the MySQL database, when a message is sent to the bot, the text will be routed to our webhook or backend for processing the message (as detailed out in our Design Document).

- Given that our Natural Language Processing API i.e. Recast.ai is set up successfully, when the user sends a message to the chat bot, then the user intent should be sent to the backend to receive a reply as 'echo message'.

2.4 User Story #4

As a first-time user, I would like the bot to repeat instructions when asking for help.

Task	Time(hrs)	Assigned To
Make Recast.ai understand that the user's intent of message is help.	6	Kush
Construct a helpful reply message.	4	Kush
Display appropriate message whenever help intent is detected.	4	Kush

2.4.1 Acceptance Criteria

- Given that Recast.ai is setup, when the user asks for help, then our algorithm should be able to understand that the user is asking for help.
- Given that the bot is able to communicate with the user, then a set of messages should be sent to the user that give detailed information about the bot's functionality and usage.
- Given that the help function is set up, when the user asks for help, then the chatbot should call the help method and render a set of messages that give information about usage and functionality of the bot at any point of time during the conversation.

2.5 User Story #5

As a user, I would like to facilitate a forum-like environment on our Facebook page.

Task	Time(hrs)	Assigned To
Provide a page where the user can ask any question to the developers.	4	Rahul

2.5.1 Acceptance Criteria

- Given that the user is having any trouble while using the bot, then the user should be able to post his/her questions on BoilerBot's page.

2.6 User Story #6

As a user, I want to be able to log into Facebook to use BoilerBot.

Task	Time(hrs)	Assigned To
Validate that the user is able to log into Facebook to use BoilerBot.	2	Rahul

2.6.1 Acceptance Criteria

- Given that the user wants to use bot, then the user should be able to login to the Facebook account using his/her own credentials and be able to access the bot.

2.7 User Story #7

As a user, I want the bot to be platform-independent.

Task	Time(hrs)	Assigned To
Validate that the bot works on all the web and mobile platforms.	2	Aditya

2.7.1 Acceptance Criteria

- Given that the user wants to use the bot as a mobile application, then the user should be able to successfully access all the features of the bot on that particular mobile platform.
- Given that the user wants to use the bot as a web application, then the user should be able to successfully access all the features of the bot on the web platform.

2.8 User Story #8

As a user, I expect the bot to act in character even when asked about something that the bot doesn't yet understand.

Task	Time(hrs)	Assigned To
Create a predetermined set of responses for unanticipated requests.	8	Devaunsh
Gracefully handle unrecognized messages, emoticons and attachments.	8	Devaunsh

2.8.1 Acceptance Criteria

- Given a case where unexpected query, attachment, or symbols, then respond with one of the predetermined messages.

2.9 User Story #21

As a user, I want the bot to be able to provide me with the location of dining courts at Purdue.

Task	Time(hrs)	Assigned To
Use Recast.ai to create location intents.	4	Eehita
Connect with Dining Court API to retrieve relevant information.	10	Rahul
Make interactive cards to show dining court pictures.	6	Aditya
Add subtext to the cards to display relevant information.	2	Eehita

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 established a connection with the Dining Court API, when a user requests location, then the interactive cards are displayed.
- Given that our database is setup, when a user requests the location of a dining court, then requested location can be added to the card accordingly.

3 Remaining Backlog

3.0.1 Functional Requirements

USER:

1. As a user, I would like to adjust the settings across the conversation.
2. As a user, I expect the bot to keep a log of all the messages across the conversation.
3. As a first-time user, I would like to have a tutorial on how the bot works.
4. As a user, I would like the bot to reply quickly.
5. As a user, I expect the bot to inform me about server downtime.
6. As a user, I expect the bot to be versatile.
7. As a user, I would like to get information regarding the last crime committed at Purdue.
8. As a user, I would like to get monthly crime statistics for the current year at Purdue.
9. As a user, I would like to know about any active warrants at Purdue.
10. As a user, I would like to get yearly crime statistics for the current year at Purdue.
11. As a user, I would like bot to provide me with information about the opening and closing times of dining courts at Purdue.
12. As a user, I would like the bot to provide me with information about food at dining courts at Purdue.
13. As a user, I want the bot to have information about the courses being offered in the current semester.
14. As a student at Purdue, I would like to know the seats left in a course that I want to register for.
15. As a student at Purdue, I would like the bot to provide me with information about the different professors teaching a course.
16. 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.
17. 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.
18. As a student at Purdue, I would like the bot to give me details like the credits offered for taking a course.
19. 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 come up with regular updates to the chatbot. Updates that include giving more relevant information based on user feedback.
2. As a developer, I would like my bot to handle at least 100 users at once.
3. As a developer, I would reduce the downtime for maintenance to 2 hours per month.
4. As a developer, I would like to be able to handle the UI for mobile and web appropriately.
5. As a developer, I would like to implement further fixes and enhancements according to the usage of the user.
6. 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.