

BoilerBot

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

January 27th, 2017

Product Backlog

Contents

1	Team 1 Members	3
2	Problem Statement	3
3	Background Information	3
4	Functional Requirements	4
5	Non-Functional Requirements	5

1 Team 1 Members

1. Aditya Dhingra
2. Rahul Pai
3. **Eehita Parameswaran (Team Leader)**
4. Kush Rustagi
5. Devaunsh Sambhav
6. Yash Shiroya

2 Problem Statement

Planning ahead for a semester and staying ahead of the game is extremely important for Purdue students. We intend to build **BoilerBot**, a Facebook Messenger chat bot that answers course related questions and efficiently provides students various Purdue resources at their fingertips. BoilerBot uniquely provides an aggregation of Purdue services by giving users quick access to useful information in one window. Thus, the bot will serve as an everyday companion to enhance the student experience at Purdue.

3 Background Information

- Balancing multiple applications for different functionality on a daily basis is a tedious task for Purdue students. Accessing information about courses, dining court menus, crime statistics on different portals increases memory usage on the individual's device and incurs a toll on battery life.
- There are several service aggregator chat bots that help people on a daily basis. For instance, Google Home has doubled up to be a personal assistant keeping the user informed about news, weather and other Google related features. Even Alexa has similar functionality with Amazon. More specifically, these bots integrate only features available through a single platform. We intend to design a chat bot that can integrate various features for Purdue students.
- Our chatbot will allow Facebook Messenger users to access these different resources through one window. Using services like *Unitime* to check for courses or browsing through the Purdue Police website is inefficient and usually can't be done on the go.
- BoilerBot is different than it's competition since it provides a unique compilation of amenities for users. Our chatbot further incorporates a simple interface to communicate with people, making the experience more personal and realistic to the user. Also, since BoilerBot uses Messenger, it is platform independent.

4 Functional Requirements

USER:

1. As a first-time user, I would like the bot to introduce itself and provide me a set of instructions.
2. As a user, I would like the bot to exchange pleasantries with me.
3. As a user, I would like to interact with the bot.
4. As a first-time user, I would like the bot to repeat instructions when asking for help.
5. As a user, I would like to facilitate a forum-like environment on our Facebook page.
6. As a user, I want to login to Facebook to use BoilerBot.
7. As a user, I want the bot to be platform-independent.
8. As a user, I expect the bot to act in character even when asked about something that the bot doesn't yet understand.
9. As a user, I would like to adjust the settings across the conversation.
10. As a user, I expect the bot to keep a log of all the messages across the conversation.
11. As a first-time user, I would like to have a tutorial on how the bot works.
12. As a user, I would like the bot to reply quickly.
13. As a user, I expect the bot to inform me about server downtime.
14. As a user, I expect the bot to be versatile.
15. As a user, I would like to get information regarding the last crime committed at Purdue.
16. As a user, I would like to get monthly crime statistics for the current year at Purdue.
17. As a user, I would like to know about any active warrants at Purdue.
18. As a user, I would like to get yearly crime statistics for the current year at Purdue. item
19. As a user, I would like bot to provide me with information about the opening and closing times of dining courts at Purdue.
20. As a user, I would like the bot to provide me with information about food at dining courts at Purdue.
21. As a user, I want the bot to be able to provide me with the location of dining courts at Purdue.
22. As a user, I want the bot to have information about the courses being offered in the current semester.
23. As a student at Purdue, I would like to know the seats left in a course that I want to register for.

24. As a student at Purdue, I would like the bot to provide me with information about the different professors teaching a course.
25. 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.
26. 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.
27. As a student at Purdue, I would like the bot to give me details like the credits offered for taking a course.
28. 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.

5 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.