

SPECIALIZATION PROJECT FINAL REPORT: DIGITAL NUDGE CHAT BOT

A Thesis

Presented to the Faculty of the Graduate School
of Cornell University

in Partial Fulfillment of the Requirements for the Degree of
MS.

by

Ta-Wei Mao, Siqi Ouyang

May 2020

© 2020 Ta-Wei Mao, Siqu Ouyang

ALL RIGHTS RESERVED

ABSTRACT

Life is full of choices and distractions especially in digital environments nowadays. It's extremely common for individuals to interact with different applications to handle their daily tasks and jobs, such as making schedules or decisions based on different information. All these behaviors require large amount of decision making. Nowadays, it's getting harder to focus on a particular task given the fact that mobile applications are distracting. How we could help individuals make better decisions and focus on a specific task are what we are interested in.

In particular, we are considering the effects of nudges in digital choice environments. To narrow down the scope, we want to leverage the capabilities of bots that are programmed to interact with users automatically. We will be focusing on simple tasks including making weekly plans, writing online documents such as Google Doc, and solving coding problems on LeetCode where people - mostly software engineers - practice interview questions for software engineer positions. We have developed three different bots for these three particular activities. These bots will have interactions with users and send nudges to them.

We will see how these chat bots would influence people's behavior in these specific activities and see how different designs would affect people's motivation and productivity.

CHAPTER 1

INTRODUCTION

Life is full of choices and distractions especially in digital environments nowadays. It's becoming crucial to figure out how to increase individual's productivity such as making more schedules and to focus more and become more productive without being distracted easily[10].

Our goal is to find effective ways to help individuals increase their productivity with the help of technologies in a data-driven way. It's known that what is chosen often depends upon how the choice is presented. In addition, the timing of sending nudges and notifications would also be important. Therefore, finding ways of building effective nudges is important for us[8]. To achieve this, building up data-driven nudges might be an answer to our research question. When to push notifications or nudge individuals so that they could have higher possibility to schedule plans? How and what tools, formats, or medium should we use to have better effects on changing individuals' behaviors? And most importantly, to build up data-driven nudges [11], what features should we collect so that we could build up the model? These are the questions that should be figured out. With this motivation, we think that the best way to verify these assumptions is to develop chat bots (here we focus on Slack chat bots) and design the nudges to test with users so that we could collect behavior data to design better nudges in future.

Within a year, we have developed three bots that help users focus on different activities: writing online documents such as Google Doc, focusing their short and weekly tasks , and solving coding problems from LeetCode which is a website where people practice their coding skills for software engineer position

interviews. There are already many bots that are existed to try to help individuals focus on some specific tasks. Some of the design and concepts of our bots were inspired by these related work which we will discuss further.

A note about this report: it is the final report for the Specialization Project, required for the Connective Media Master program. This project was a two-person research project done under the advice of a faculty member at Cornell Tech.

CHAPTER 2

RELATED WORK

We did some research about nudging and it seems to have a great effect in human[11]. Then we looked up some specific paper that talk about or use chat-bots to nudge people. With the growing sophistication of robots, it is becoming increasingly likely that the technology will have a profound influence on human behavior. Equipped with the knowledge of affective computing and other disciplines, roboticists could exert profound power over the humans that interact with robots[6]. And Researchers use artificially intelligent bot programs to stimulate collaboration and make people more effective[7]. There are already many bots that are existed to try to help individuals focus on some specific tasks. We have compared several ones that want to achieve similar goals and figured out the design ideas for our bots. Below we will list the most related bots.

- MakePlans[3]:

MakePlans is an application for appointment booking, scheduling and class/event sign ups. For every new appointment that is created in MakePlans this bot will notify users in their Slack chat.

- Reminder Bot[4]:

This is a bot that reminds users for daily, weekly or monthly routines such as taking medicine, going to gym and etc. Once users have set up the chat bot, it will send users notifications to ensure that users' tasks are completed and logged over time.

- FocusTime[2]:

This application uses RescueTime to track users' activities and changes user's presence on Slack. It can help cut down on unintentional distractions from other people while users are trying to focus. It can optionally post a message to the chosen channel so everyone knows that the user shouldn't be disturbed.

- TimeCamp Timer[5]:

TimeCamp is the time tracking application designed to help teams and individuals accomplish more in less time. It lets users know how much time they spend on communication and track time spent on channels for further analyze. It also has a better progress visualization.

- Beebot[1]:

Users can send a direct message to check on their most urgent goals, see what they need to do to meet them, and add data once users have completed them. The tool also comes with a slash command that lets users do all the same things without leaving the channel users are in.

From the above competitive analysis, we see that there are separate applications focusing on different components such as making plans, setting goals, and etc. However, there are no existing chat bots that combine these features together. Based on what we have discovered, we decided that our chat bot should build

features all in one including making weekly plans, helping users stay focus in a short time, and avoiding distractions. In addition, we will also incorporate external data sources such as RescueTime and Google Calendar to better understand users' activities and habits. Besides, we planned to focus on making users' interaction and experience more natural and intuitive, enabling users to interact with our chat bot by simply sending direct messages instead of having to use lots of slash commands. In addition to focusing on task productivity, there are no existing bots that help users keep on track of their online documents writing progress. There are no other bots that help users focus on solving LeetCode problems either.

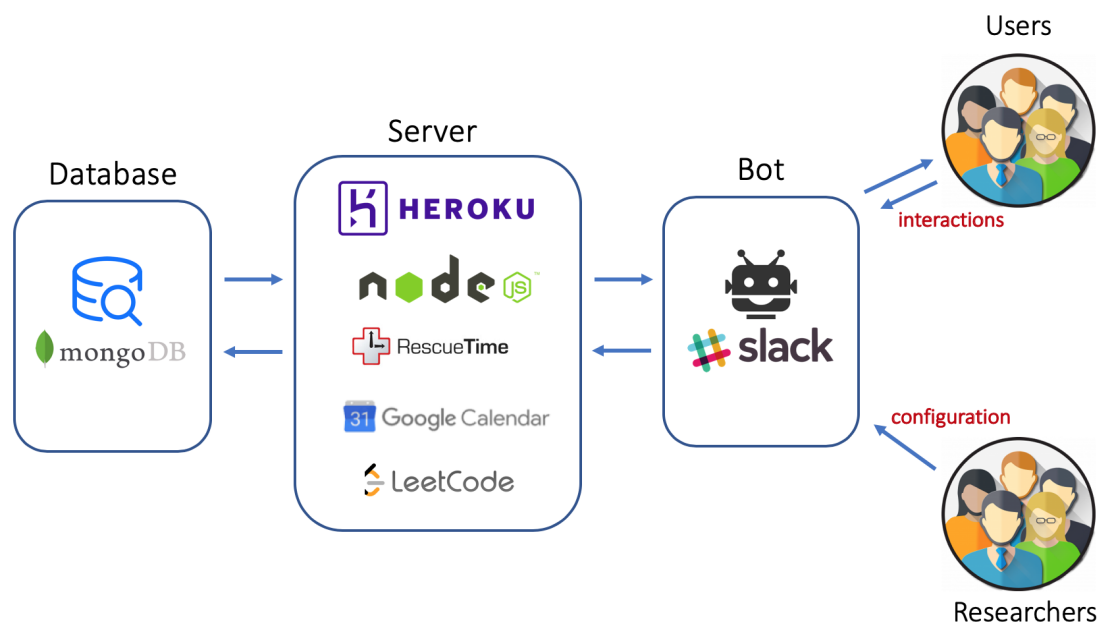
CHAPTER 3

IMPLEMENTATION

Our chat bot is built in the Slack ecosystem. For our technical approaches, we used a open sourced library called "slackbots" to build up our chat bot. The main programming language used is JavaScript and our backend server is NodeJS. As for our database, we use MongoDB to store users data such as RescueTime tokens, Google Calendar Token, Slack token, and LeetCode account token. We only store tokens and user activities on our chat bot instead of other private user information such as their messages on Slack. Our system is deployed on Heroku, a PaaS platform that enables us to build, run, and operate the application entirely in the cloud.

The diagram below is an overview of our system architecture. For research purpose, our bots allow researchers to upload configuration settings directly

through the bot. The configuration settings allow researchers to change things like messages shown to the users, time settings, and etc. Therefore, researchers won't have to dive into the code to change the behaviors of the bot, making it easy to conduct AB testing and other investigations. For our three bots, we deployed each of them on different Heroku servers, calling different APIs and third-party services so that our bots will conduct different tasks and interactions with our users. But overall, this structure is the same for all our three bots.



- Software Components:

In order to have comprehensive functionality, we Incorporated several API endpoints provided by several external parties[9]:

- Slack APIs:

Slack APIs allow us to better keep track of users' activities on Slack as well as improving the bot's functionality and build up a more intuitive user experience. The Slack APIs allows us our chat bot to send messages, create simple workflows such as responding to user ac-

tivity and generating buttons to let users complete simple tasks like requests and approvals.

– RescueTime APIs:

RescueTime APIs help us keep track of users' activities such as how much time each individual has spent on each categories including software development, design, learning new things, and more. The APIs we used are Analytic Data API and Daily Summary Feed API. With Analytic Data API, we could see how many hours each individual has spent on each category in which specific time ranges. As for Daily Summary Feed API, we could see a high level roll up of the the time a user has logged for a full 24 hour period.

– LeetCode:

LeetCode doesn't provide a comprehensive API that allows us to get user's information such as problem submissions etc. Therefore, we used JavaScript scraping libraries to scrape information displayed on the website.

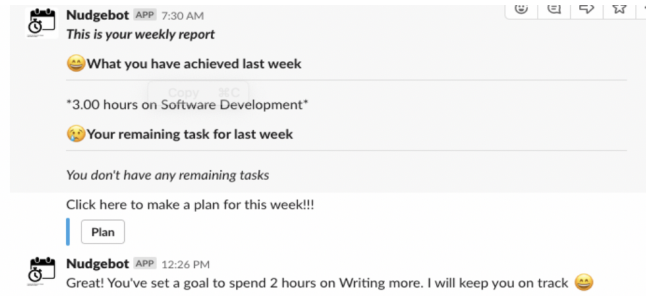
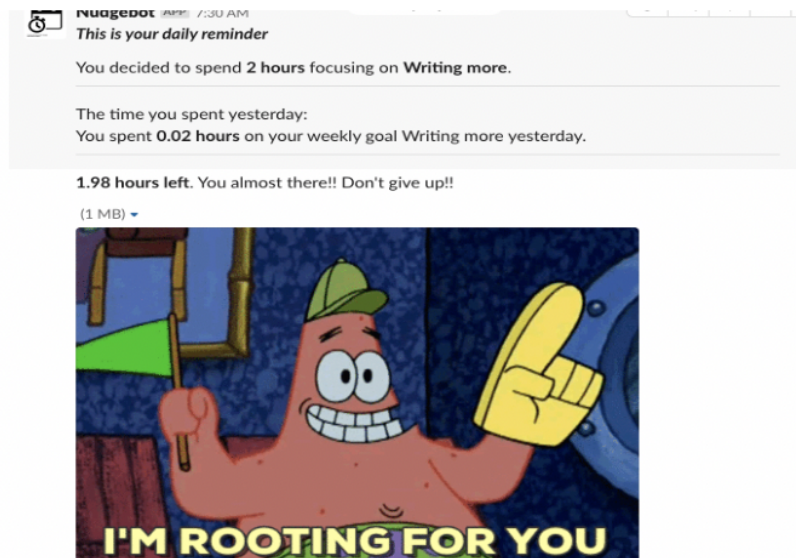
• Client-Side and Server-Side:

We first built the chat bot on Slack with basic functions such as connecting to individual's Google Calendar. We've also connected to each individual's RescueTime to better keep track of users' activities and use the APIs from RescueTime to build features in our chat bot. In addition, we've designed the message formats that could be easily triggered by users so that no too many instructions needed. After we have built up the basic functionality and core features, we then conducted several AB testing to compare and investigate how different messages or message formats sent by chat bot would affect individual's decisions. Also, we built up a new

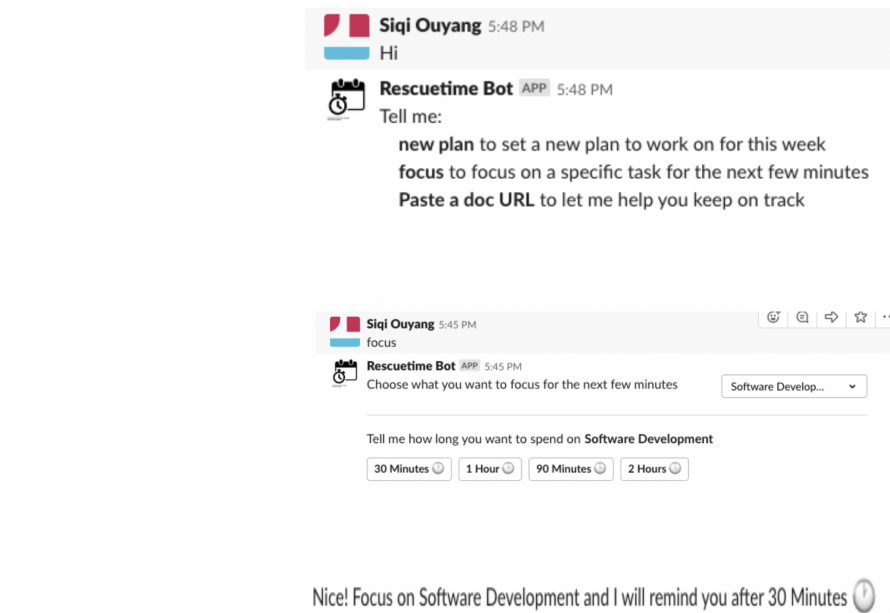
Slack work space and recruited participants to test our chat bot, gathering user feedback to improve and optimize our chat bot in the future. With these data collected, we could then analyze the factors that influence user behavior.

- Slack: With the connection to Slack, our bot could analyze how long each individual has spent on Slack and we could know users' activities in general. With this data, in future we could develop and design the best notifications that could nudge users at the right time.
 - RescueTime: With the connection to RescueTime, as mentioned before we could better keep track of users' activities and use the APIs from RescueTime to see how many hours each individual has spent on each category in which specific time ranges.
 - Google Calendar: With the connection to Google Calendar, we could see each individual's distribution of their schedules. For example, we could know whether the user is more likely to work in the mornings or at night. We can not know the details of users' events and schedules, so there are no privacy issues.
- Feature Components:
 - Weekly goals: Every Monday, users will receive a weekly report to summarize their progress and goals that were achieved and not achieved. Then the bot would ask the user to set a weekly goal they want to focus on. There are three options provided including software development, learning new things, and writing. Users have to decide how many hours they want to spend on the goal before the week ends. Once users have achieved the goal, they can make

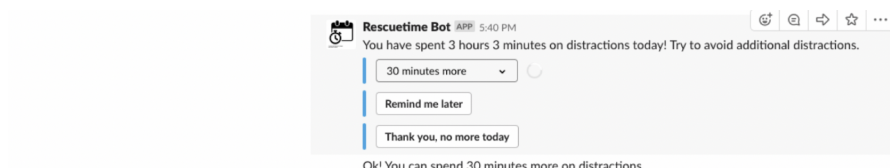
another goal for the rest of the week. From Tuesday to Sunday, users will receive daily reminders in the morning, telling users their progress and how well they are doing. The reminders include GIFs to encourage users. We use RescueTime to get users' activities data to know how much time each individual has spent on the activity.

A dialog box titled 'New FOCUS for this week!' with a close button (X) in the top right corner. The text inside says 'Select one thing you want to focus for this week' followed by a dropdown menu showing 'Writing more'. Below that, it asks 'How many hours do you want to spend' with an input field containing the number '2'. At the bottom, there is a link 'Learn more about Rescuetime Bot', a 'Cancel' button, and a green 'Request' button.

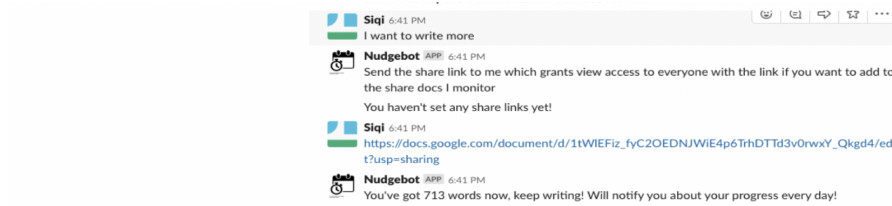
- Short focus: Besides weekly goal, users can set a short time period to focus on one activity which is software development, learning new things, or writing. The time options are 30 minutes, 60 minutes, 90 minutes, and 120 minutes. After the time has passed, the bot would check whether the user has fully focused on the task by using RescueTime data.



- Distraction: In addition to enabling users to choose what to focus on and make weekly plans, our bot would also calculate how long each individual has currently been distracted. We also use RescueTime to get the information. The bot would ask users how long they want to continue spent time on other stuff instead of software development, learning new things, and writing. The messages sent to users varies based on users' activities.



- Online Document Progress Tracking: Users can send online documents such as Google docs, txt, doc file from Dropbox, or original Overleaf link to the bot. And the bot would check whether users are making progress working on their documents based on the word counts.



- LeetCode bot: In order to get users' activities on LeetCode, we need users' account token. Unfortunately, LeetCode doesn't provide OAuth. Due to this, we require our users to go to their browser and copy their cookies to our bot. We emphasize that we would only use their cookies to access their activities on LeetCode such as recording number of problems solved, etc. We developed features such as setting weekly goals of how many LeetCode problems users plan to solve for that week, daily reminders reminding users to take actions to meet their goals, and a leader board where users can see other users' progresses. All these implementations are deployed from the same concepts mentioned earlier.

CHAPTER 4

RESULTS

We have recruited 5 participants in total to try the productivity bot. Only 2 participants have done it and provided feedback. As for LeetCode bot, we have

recruited 12 participants to test and see if the bot would really motivate them more to solve more LeetCode problems.

- Productivity bot: The overall experience is not good for both users. One of the users says that he will not open the bot in Slack to tell it what he is going to focus on for the next few minutes. He gets more distracted comparing to not using the bot. As for the features, they both think that the distraction feature is confusing and disturbing due to high-frequency notifications from the bot. Besides, one of the users thinks that the onboarding process is too complicated and requires lots of procedures. One suggests that the bot should incorporate social validations such as a leader board to compete focus time.

- LeetCode bot: Comparing to the productivity bot, users have more motivation using the LeetCode bot, since the more problems users have solved, the higher chance they will do better at their software engineer interviews.

We assume that with LeetCode bot, users will solve more problems given that users have to set their own weekly goals along with daily notifications reminding users to work on track. However, from our testers' feedback, we found out that most users feel pressured, resulting in few problems solved. We then conducted another shorter tests, telling users that the bot will be their "assistant" instead of "supervisor". With this emphasis, surprisingly, the number of solved problems increased a little bit more.

In addition, we also assumed that with the concept of social validation such as leader board where users can see other users' progress, they will be more motivated and end up more problems solved. However, it turns out that there's no too much difference. And some users are even feeling

stress more than without the leader board.

A good suggestion from all our testers is that we should provide default options for setting their weekly goals of solving LeetCode problems, such as providing three options including 3, 5, or 10 problems for the week. This could help users to make less decisions and be more engaged.

CHAPTER 5

DISCUSSION

The goal of the project is to find and develop the most effective ways to help individuals focus and become more productive on specific tasks. We developed three bots that help individuals focus, write online documents, and solve more LeetCode problems. We applied the concepts as well as similar designs from other existing bots to our implementation. From the results and users feedback, however, we discovered some interesting facts and things that we didn't well consider before. Here we will point out some of them so that these could be emphasized more in future for researchers looking to design such a bot or students that want to develop similar tools.

5.1 Limitations

One of the biggest pain points of using the LeetCode bot is that we require users to get their cookie manually, which causes confusions and makes users wonder whether their privacy would be violated. This actually not only happens to LeetCode bot but also productivity bots. The acquirement of user identities

including token and cookies would make users concerned. In addition, this could discourage users from using the bots to increase their productivity. This should be addressed more for future developments in bots. Developers should also take more serious and careful considerations when using third party tools to monitor users' activities.

In addition to user tokens, our bots also rely on third-party applications such as RescueTime, which require our users to install on their computers. If RescueTime is not installed on users' computers, their activities will not be able to be tracked for the bots to function as we wanted. In addition, if users use their mobile phones more than using their computers, our bots will also not be able to track users' activities precisely, since all the activity data are collected from users' computers. Researchers and developers will need to be aware of this in future for further research.

5.2 Lessons Learned

From the three bots we have developed and testers' feedback, we see that users did feel more productive. However, the effect doesn't last long. After using a few days, their progress of working on their goals slowed down. Moreover, some users are still aware of their privacy issues even though they were well informed that their privacy and activity data would not be violated. These concerns discourage them to use the bots and weaken their motivations.

Another insight we have is that in general, the existence of these bots would stress users more since most users feel that their activities are being monitored, which would affect their motivations and performance as well. Therefore, how

we should inform users about what these bots would and could do is important.

In conclusion, for future bot developments, we shouldn't just focus on the development. We should also be aware of how users' behaviors. It is also important how we frame the messages to users, emphasizing that these bots would only be their assistants. And sometimes social pressure and validation will not work. The purpose of these bots should be motivating users and help them become more productive. If we only focus on bot development and ignore these important factors, this could backfire and discourage users.

5.3 Future work

For future work, we should improve the overall pipeline based on users' feedback and further research. After that, we will deploy the pipeline to other platforms such as Facebook Messenger. In addition to incorporating RescueTime, we will also add more data sources such as Apple Health to develop more features. Most of all, we should put more emphasis on users' behaviors and feedback to modify the concept behind the nudges and the bots.

CHAPTER 6 CONCLUSION

Nowadays, it's getting harder to focus on a particular task given the fact that mobile applications are distracting. Developing a digital assistant to help individuals focus and increase productivity is drawing more and more attention. In particular, developing a bot could be one of the solutions. However, researchers

and developers should not only just focus on development. There are other issues that need to be considered. We should be aware that a bot will not assist individuals focus more and increase more productivity in every activity. Activities with more specific goals such as solving LeetCode problems would be more effective with the bot. In addition, we should also be careful about how the bot frames its messages to users. If researchers and developers only focus on the bot development, it could backfire enormously.

CHAPTER 7

ACKNOWLEDGEMENTS

This work would not have been possible without the full support from Fabian Okeke, PhD student at Cornell Tech, and Michael Sobolev, postdoctoral associate at Cornell Tech. They provided insight and expertise that greatly assisted the project.

BIBLIOGRAPHY

- [1] Beeminder. <https://www.beeminder.com/home>.
- [2] Focustimeapp. <http://focustimeapp.com/>.
- [3] Makeplans. <https://makeplans.net/en/>.
- [4] Makeplans. <https://reminderbot.io/>.
- [5] Timecamp. <https://www.timecamp.com/>.
- [6] Jason Borenstein and Ronald C Arkin. Nudging for good: robots and the ethical appropriateness of nurturing empathy and charitable behavior. *Ai & Society*, 32(4):499–507, 2017.

- [7] Richard Conniff. Pushy ai bots nudge humans to change behavior, May 2017.
- [8] Uri Gneezy, Stephan Meier, and Pedro Rey-Biel. When and why incentives (don't) work to modify behavior. *Journal of Economic Perspectives*, 25(4):191–210, 2011.
- [9] Keith Horwood. Build a “serverless” slack bot in 9 minutes with node.js and standard library.
- [10] Katherine L Milkman, Dolly Chugh, and Max H Bazerman. How can decision making be improved? *Perspectives on psychological science*, 4(4):379–383, 2009.
- [11] Cass R Sunstein. Nudging: a very short guide. *Journal of Consumer Policy*, 37(4):583–588, 2014.