

Individual Report

CSE 5911

Instructor: Perumal N. Ramasamy

Sponsor: Branden Bellanca

Team iBot

Xiao Liang

Xuzhou Yin(myself)

Young Liu

Yun Ma

1. Background

With the rapid development of technology these days, projects and research become more and more complicated. It already takes so much time for IT staff to design and build their technical projects. However, some tasks they do every day require cumbersome steps and unnecessary complexity. Therefore, they do need a product to help them with it. This project our team work on is a chat assistant which aims to solve this problem. The project, which is called iBot, is an intelligent bot which has Natural Language Processing Services integrated and interacts with issue and project management tools such as Jira, and continuous integration tools like Jenkins so that it helps turn inefficient steps into dialogs to increase productivity for all employees. Currently, iBot supports channels like Skype, Slack and Facebook Messenger. The project management tool it supports is Jira. In future, iBot will support more and more DevOps tools such as BitBucket and Jenkins.

2. Individual Contribution

Since my roles for this project were secondary Project Manager, primary Test Manager and secondary Documentation Manager. I involved into designing, coding, debugging and documentations. Followings are my specific distributions for each part.

2.1 Sign in, sign out and switch accounts

I initially considered to make the sign in page a webpage, so that user would receive a link to the page through iBot. To do this, I did research on NodeJS webpage techniques such as express.

However, I then realized that it would be better for user to login just through iBot itself instead of visiting a page through browser. So, I made the feature that user can login with their own Jira account in iBot chatting interface. User can type “LOGIN” to sign in, “LOGOUT” to temporarily sign out, and “SWITCH” to switch to another account.

Account feature consists three parts. The first part is for signing in. I disabled all the commands before user signs in so that user is not able to do anything before he or she is signed in. To sign in, iBot will ask user for Jira username, password and Jira instance URL. The reason why it asks for Jira instance URL is that inside the internal functionalities of iBot, it uses RESTful API to interact with Jira. Therefore, for each operation such as create, modify and search, iBot will need a URL link. Since each Jira instance has different URL, so that iBot needs to get the specific URL from user to apply to the further operations.

The second part is for temporary signing out. If user need to be temporarily away, and he or she does not want others to use his or her account, iBot makes the user be able to sign out. However, in this situation, iBot remembers user’s account information. Thus, when the user comes back and tried to login, iBot will only ask the user for password. The account information is stored in the program cookie.

The last part is for switching account. I considered the situation that a user is involved into multiple groups and working on multiple projects. Now, user can switch to his or her another account through iBot. So, when user switches to another account, iBot will clean all the account information and ask user for username, password and jira URL of new account.

2.2 Jira authentication

I am also responsible for the Jira account authentication part. Since iBot uses RESTful API, it must be authorized by Jira. At that time, there were three choices for me to choose: OAuth, Cookie-based and Basic authentication. At beginning, I adopted and implemented OAuth authentication in JavaScript. However, after discussing with other group members, I decided to re-implement it by using another authentication method. The reason is that the one I implemented uses OAuth1, which has some security bugs. However, Jira only accepts its own applications (not third-party applications) use OAuth2. After discussion between our team and sponsor, we decided to implement it in Basic Authentication method. After that, I re-implement it with basic authentication integrated with the sign in feature as described above.

2.3 Testing and debugging

During the development of this project, I helped another team member of our group with testing. With the new feature came out, I involved into the debugging process. I fixed several bugs that were reported by other team members. For instance, iBot had a bug that if user initially tried to create a story, and then canceled the action by typing “CANCEL” when iBot asks for story summary, next time no matter what kind of task user choose to create, iBot just regarded it as story. It was the problem of the functionality for “CANCEL” action. Thus, I rewrote the “CANCEL” function and got the bug fixed.

2.4 Documentations

I involved into the documentation part throughout the project. I cooperated with my teammates on the initial problem statement and project charter. In addition, I participated the group discussions and design the presentation PowerPoints for weekly standups, risk presentation, meetings with sponsor and the meeting with VP. At last, I also helped with the design and content for the project poster.

2.5 Class participation

I participated to all the lectures and classes without any absence. I did not miss any class and weekly standup. In addition, I participated the majority meetings with sponsor and took notes. There was only once I missed the meeting with our sponsor, and this was because I had a class which I cannot skip. Also, I went to all the group meetings and join the projects discussions including design, workflow and presentation preparation.

3. Technologies Learned

3.1 Node.js framework

Before I took this course, I have only heard about Node.js framework. The only thing I knew about is that it uses JavaScript. However, as the development of this project, I feel like Node.js is much more powerful than I originally thought. For example, we can use Node.js to build any

kind of server. For instance, Microsoft Bot Framework server is built on the top of Node.js. In addition, not only JavaScript, but also TypeScript can be used in Node.js.

3.2 TypeScript

As a superset of JavaScript, TypeScript supports strong typing. Before our team started this project, I had not even heard about TypeScript. However, I needed strong typing so that I wouldn't get confused when calling functions. Throughout the process of this project, I learned a lot about TypeScript including the syntax and the concept of strong typing.

3.3 Microsoft Bot Framework

Another technology I have learned is Microsoft Bot Framework. This is a pretty new framework developed by Microsoft. This framework can be hosted to server (our team current hosts it to Heroku Web Services). And it supports multiple channels such as Skype, Slack and Facebook Messenger. It simulates a chatting assistant and answer user's questions. To allow some typos, it supports another product called LUIS, which is also made by Microsoft.

3.4 LUIS (Language Understanding Intelligent Service)

As another incredible product designed by Microsoft, LUIS did a nice job to help bot framework deal with the user input by Natural Language Processing. Through the process of this project, all team members learned how to define intents and other similar syntax through LUIS. In addition, all the members also become familiar with training LUIS to make it "smarter". With the help of LUIS, iBot becomes able to bear some typos of users and accurately recognize the intents from user.

3.5 Authentication

Authentication is also one of the main technologies I have learned. Before our team started this project, we all thought authentication is easy and it could be done very quickly. However, after playing along with it for about one month, we changed our mind. We realized that authentication heavily depends on specific projects. For instance, OAuth2 is the most secure way of doing authentication. But Jira does not allow third-party applications to use OAuth2. OAuth1 has bugs and is not secure. Thus, our team finally chooses Basic Authentication since it is relatively secure and does not require much work done.

3.6 Agile Methodology

Since I did not take any course for Software Engineering specialization, I knew nothing about software development methodologies like Agile. However, through this semester, I learned a lot about it, including what Agile is, how it works and why we need this. Thanks to Agile, I feel the efficiency of our project increases rapidly and all team members are much clearer about the progress of the project.

4. Roles

4.1 Xiao Liang

Xiao was the overall project manager for our team. He contributed a lot to the project, including choosing proper framework, making weekly plans and many technologies. Other than that, he

was also primary designer and secondary code manager. He was more familiar with Node.js framework so he taught us how to write code under this framework.

4.2 Xuzhou Yin

My roles for this project were secondary Project Manager, primary Test Manager and secondary Documentation Manager. I involved into designing, coding, debugging and documentations.

4.3 Young Liu

Young was Primary code manager and secondary test manager. His contribution focused on coding, testing and documentations.

4.4 Yun Ma

Yun was secondary design manager and primary test manager. Her contributes a lot on designing, testing and documentations.

5. Peer Evaluation

Overall, I had a happy time spent as a part of our team. And I believe everyone in our group has even distributions to this project. And I enjoy the development process of this project and I am happy to learn such a bog number of things. At last, I would like to evaluate all my team members excellent.

6. Discussion and Methodology

Other than the class time, our team met almost twice a week (sometimes Tuesday afternoon, sometimes Weekends, sometimes both). And we had meeting with our sponsor almost every week. The methodology we used is Agile. And the platform we use to track the progress of our project is Jira. We completed and created new sprints every two weeks. The version control system we used is git. And the platform we used to share and cooperate is GitHub.

7. Conclusion

In conclusion, this course has brought me the most experience and helped me prepare for the real-life project I will encounter in future. I enjoy this course not only because I learned lots of technologies, but also the strategies for developing software and even getting along with others. All in all, this is the most valuable course I have ever taken in my life. And I will remember the time I spent with my team forever!