Book Bot Project

Group 7

October 27, 2017

Abstract

This file is a project report about Group 7's Book Bot Project for CmpE451 class. Information in this file is up to Milestone 1.

1 Project Description

Book Bot Project is an assistant that can help users with everything related to books. It is a bot, and it works via Telegram. Working via Telegram makes the bot more accessible; users don't need to install anything extra and it is cross platform.

The base and most important functionality is to communicate with natural language. The bot can carry on a conversation with daily English.

Users can ask the bot about books. They can search books with different keywords, genres or authors, they can filter them in any way they want and they can sort them. They also can get information about specific books, see its current popularity via ratings or read the comments about the book.

Users can also give the bot feedback about the books they have read. They can comment on a book or rate a book. This will help the bot learn more about what the user likes and help it work more user oriented

The bot can also recommend the user some books depending on their previous ratings and taste of books.

1.1 Project Requirements

We have defined our project requirements as follows and our customer has agreed and approved the requirements. Note that the following is a summary of the detailed requirements page. You can see the full list from project requirements in our wiki page.

1. Functional Requirements
   (a) System Requirements
      i. Telegram will be the medium for the bot.
      ii. Goodreads or Google Books will be our source of book related information.
      iii. Wit.ai will be used for NLP purposes to understand natural language.
      iv. There will be regular users, admins and moderators.
      v. There will be a conversation tree which is going to be used to determine the conversation direction.
   (b) User Requirements
      i. We will have regular users as Telegram Users.
         A. Users will be able to get information; search for books, filter them, sort them and read comments about them
         B. Users will be able to give feedback about books; comment on them and rate them.

        C. Users will be able to get book recommendations.

    ii. We will have admin users.

        A. Admins will be able to modify the conversation tree
        B. Admins will be able to see and flag comments
        C. Admins will be able to see and block users

    iii. We will have moderator users.

        A. Moderators will be able to see and flag comments
        B. Moderators will be able to see and block users

2. Non Functional Requirements

    (a) Speed: We are creating a bot which users can chat with so the response time cannot be more than 5 seconds.

    (b) Size: We want our bot to be portable an not bulky, we will use the Telegram as our medium hence the user won't need extra space for the bot.

    (c) Usage: It should be easy to use, there shouldn't be any training time.

    (d) Requirements: Bot needs internet connection to work.

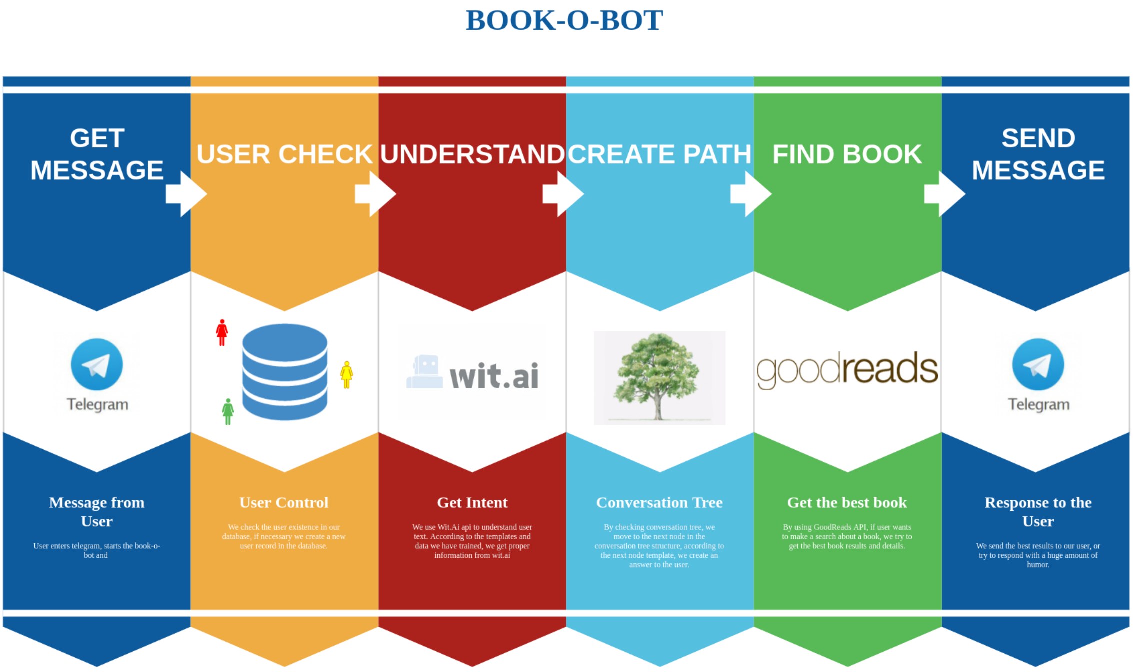
    (e) Failures: If bot fails, it shouldn't take more than ten minutes to restart the system and get it working again.

    (f) Target devices: We are targeting all platforms in which Telegram works.

    (g) User Satisfaction: Bot should keep track of the success of its recommendations by tracking the ratings of the recommended books.

1.2 Project Design

See Figure 1



BOOK-O-BOT

GET MESSAGE

Message from User

User enters telegram, starts the book-o-bot and

USER CHECK

User Control

We check the user existence in our database, if necessary we create a new user record in the database.

UNDERSTAND

Get Intent

We use Wit.Ai api to understand user text. According to the templates and data we have trained, we get proper information from wit.ai

CREATE PATH

Conversation Tree

By checking conversation tree, we move to the next node in the conversation tree structure, according to the next node template, we create an answer to the user.

FIND BOOK

Get the best book

By using GoodReads API, if user wants to make a search about a book, we try to get the best book results and details.

SEND MESSAGE

Response to the User

We send the best results to our user, or try to respond with a huge amount of humor.

Figure 1: Project Design

Row Name	Row Name
Column Name	column value
Column Name	column value

Table 1: An project plan table.

1.3 Project plan

TODO: write project plan. It is going to be in tabular form. See Table 1

2 Project Milestones

2.1 Milestone 1

Book bot works fine for getting information purposes. (End of October)

2.2 Milestone 2

Book bot works fine for giving information purposes. (End of November)

2.3 Final Milestone

Book bot is able to recommend books. (Project is finished)

3 Project status

3.1 Deliverable List

1. Goodreads API
2. Telegram API
3. Node/Conversation Graph
4. Database Creation
5. Cloud
6. Telegram Service(Cloud)
7. Sign-up/Login
8. Settings
9. Admin Dashboard
10. Viewing Conversation Graph
11. Testing Demo

3.2 Deliverable Status

TODO: write deliverable status. It is going to be in tabular form. See Table 2

Row Name	Row Name
Goodreads API	Delivered
Telegram API	Delivered
Node/Conversation Graph	Delivered
Database Creation	Delivered
Cloud(AWS)	Partially Delivered
Sign-up/Login	Partially Delivered
Settings	Not Delivered
Admin Dashboard	Partially Delivered
Viewing Conversation Graph	Delivered
Testing Demo	Partially Delivered

Table 2: An deliverable status table.

3.3 Deliverable Evaluation

1. Goodreads API
   (a) It is delivered however we may change this API to Google Books due to latency problems.
2. Telegram API
   (a) It is done and since it was the main thing to communicate with users, it was crucial. We build whole structure onto it.
3. Node/Conversation Graph
   (a) We have done this, it works properly. The conversation graph is traversed while conversation.
4. Database Creation
   (a) The database models are created as designed in the design plan.
5. Cloud(AWS)
   (a) Cloud(AWS) is set up however we couldn't make it work exactly, as it is on our local.
6. Sign-up/Login
   (a) It is delivered with the help of DJango interface. We may need to customize this to our needs in the future.
7. Settings
   (a) We couldn't find energy and time to implement this part. It will be ready on the 2nd milestone. This wasn't on the critical path , so it isn't a big deal to being this deliverable is not delivered.
8. Admin Dashboard
   (a) It is delivered with the help of Mptt interface. We may need to customize this to our needs in the future.
9. Viewing Conversation Graph
   (a) It is delivered and it was necessary to see conversation trees structure.
10. Testing Demo
    (a) We couldn't tested the demo version very well, we planned to put our internal milestone 3 days before the customer milestone to prevent this happen again.

4 Coding Work

You can find each team member's contribution to the code in the Table 3

Name	Coding Work
Salih Sevgican	GoodReads API implementation
	book API class for ease of use
	Main Page of the project (Front End)
	AWS has been configured for the deployment
	Updated requirements.txt
	GoodReads compatibility with TelegramBot tested on telegramapi
Irmak Kavasoglu	Initial environment setup for Django
	Setting up database model for Telegram User
	Setting up database model for Templates
	Setting up database model for Conversation Node
	Integrating Mptt tool for conversation tree
	Registering models to admin page and customizing their look
Ali Goksu Ozkan	Implementing GET/POST to functionalize Node class
	Implementing GetNextNode class to function Conversation Tree
	Conversation Tree integration with Telegram API
	Conversation Tree compatibility with wit.ai
M. Melih Mutlu	Telegram API implementation
	Writing GET and POST methods for our API
	Integrating Telegram bot with Wit.ai
	Deploying the project to AWS
	Node change operations for Telegram User
Taha Kucukkatirci	Wit.ai API configuration
	Intents have been defined and trained with some training inputs
	Implementation of getIntent method
	Implementation of the method to auto train Wit.ai app
	Validation of the requests from users and retrain by them if valid
Add your name	add your contribution

Table 3: Coding work table.

5 Evaluation of tools and managing the project

5.1 Django

Django is a free and open-source Web framework implemented for Python. It led us writing your back-end without needing to reinvent the wheel. Since Django's primary goal is to ease the creation of complex, database-driven apps, we decided to use it. It was also in the emerging technologies according to our software tool research. It was easy to learn, install and set configuration.

Ridiculously fast.

5.2 Mptt

MPTT is a API to store hierarchical data in a database. The purpose of the tool is to make retrieval operations efficient. We needed a conversation tree in our software structure. But implementing tree structure was a little unefficient for us. We needed to visualize the tree for the admin use and we needed other functions such as get_Children. We thought that we could focus on the functional details rather performance or implementation details of the tree. Therefore we decided to use this API. The API was very useful and made our job on the conversation tree easy. What we learned from this experience is utilizing a functional and well-implemented API would make your project management better.

5.3 Telegram

Telegram offers HTTP-based interface to do most of the operations that we could need for a chat bot. Receiving and sending messages are the ones that we had to implement in first place. Telegram API provides all messages that the bot received in last 24 hours. For this reason, we had to careful to not processing same messages more than once. Response time of Telegram API is satisfying so far. As far as we read in documentation, Telegram API supports actions that we will need for this projects. However we haven't used it for sending and receiving more complicated messages such as book results that includes images, links or buttons.

5.4 Wit

Wit.ai is a service provided by Facebook for intent detection. It is a very useful tool and makes our job much easier. First of all, we created our app on Wit.ai and then defined our intents and train them with some training sentences. After enough training data, it gets an insight about the data and catches the patterns for each intent. Then, it detects the intent of the unseen, new sentences easily. Like said before, it is a very useful tool and speed up our project by handling intent detection issue. We'll also use it for further purposes like getting and remembering context of the conversation and this will make our bot answer in a more coherent and smarter way.

5.5 GoodReads

GoodReads API is very useful to make a search about a book by given any kind of keyword. Keyword can be related to author, genre or title, doesn't matter, this api makes a succesfull search every time. We've used a wrapper library for GoodReads API which made easy to write methods for search types. Library is a litlle bit complicated despite it is written to ease our job. Unfortunately not every book is being hold as a same type in goodreads database, some of queries returns arrays with additional information of book such as order number of a book in the book series, and some of queries returns just the name of book. This is an issue we need to handle very well in order to make a great impression with the responds to the bot user.

5.6 AWS

AWS is the service that we use for our project's server need. It is very convenient to start an instance and easy to use. Connection to the Amazon server is very easy on Ubuntu and our group members mostly use Ubuntu. On Ubuntu, by ssh, we can connect to the Amazon server with one line command.We first created an AWS account and then waited for approval of initial payment. Then we launched an EC2 instance on ubuntu and we are given a .pem file. We stored it in a folder and changed its privacy by chmod 400 command. Lastly, we achieved to connect server with given AWS id and ip by ssh on ubuntu. We sent the .pem file to other group members and told the process to them to make them able to connect to the server.

6 Summary

TODO fill