Project Name: Schedule Bot

SE Group 19

- Leslie - tliu33
- Yi Ting Hou - yhou9
- Deepak - dpendya

Project link: https://github.com/CSC510-Leslie-Tim-Deepak/Enigma

Rubrics

Notes	Marks/Points(0-3): Leslie	Marks/Points(0-3): Yi-Ting	Marks/Points(0-3): Deepak	Evidence
Workload is spread over the whole team (one team member is often X times more productive than the others...	3	3	3	Collaborators Commits Link
But nevertheless, here is a track record that everyone is contributing a lot)	3	3	3	Collaborators Commits Link
Number of commits	3	3	3	commits 236 in GH Leslie - 12 Yiting - 6 Deepak – 5 Command: git shortlog -s -n
Issues reports: there are many	3	3	3	0 Open Issues
Issues are being closed	3	3	3	2 Closed Issues
Docs: docs generated, format not ugly	3	3	3	Documentation
Docs: what: point descriptions of each class/function (in isolation)	3	3	3	Functionality
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	3	3	Screenshots were placed for all the features newly added and placed in the README.md doc page entries
Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	3	3	3	Elaborate documentation, detailed explanation of the functionality and how to use the bot has made it easy to understand and utilize it to its full capacity Video
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	3	3	ReadMe
Use of version control tools	3	3	3	Github
Test cases exist	3	3	3	CodeCov
Test cases are routinely executed	3	3	3	File1
Issues are discussed before they are closed	3	3	3	Discussion1 Discussion2
Chat channel: exists	3	3	3	Chat Channel
Test cases: a large proportion of the issues related to handling failing cases.	3	3	3	Test Cases
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	3	3	CheckCommits
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	3	3	Requirements.txt is installed by everyone.
Evidence that the whole team is using the same tools (e.g. tutor can ask anyone to share screen, they demonstrate the system running on their computer)	3	3	3	All the members are using the same tools
Evidence that the members of the team are working across multiple places in the code base	3	3	3	CheckCommits
Short release cycles	2	2	2	Graphs
The file .gitignore lists what files should not be saved to the repo. See examples	3	3	3	Gitignore
The file INSTALL.md lists how to install the code	3	3	3	INSTALL.md
The file LICENSE.md lists rules of usage for this repo	3	3	3	LICENSE.md
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	3	3	CODE-OF-CONDUCT.md
The file CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up; e.g. see example	3	3	3	CONTRIBUTING.md
The file README.md contains all the following	3	3	3	README.md
Video	3	3	3	Youtube Link
DOI badge: exists	3	3	3	ReadMe
Badges showing your style checkers	3	3	3	Style Checker
Badges showing your code formatters	2	2	2	Formatters
Badges showing your syntax checkers	3	3	3	Syntax Checker
Badges showing your code coverage tools	3	3	3	README.md
Other automated analysis tools	2	2	2	README.md
Q1 - What your software does				
1.1) Does your website and documentation provide a clear, high-level overview of your software?	YES	YES	YES	
1.2) Does your website and documentation clearly describe the type of user who should use your software?	YES	YES	YES	
1.3) Do you publish case studies to show how your software has been used by yourself and others?	YES	YES	YES	
Q2 - Your project's and software's identity				
2.1) Is the name of your project/software unique?	NO	NO	NO	
2.2) Is your project/software name free from trademark violations?	YES	YES	YES	
Q3 - Availability of your software				
3.1) Is your software available as a package that can be deployed without building it?	NO	NO	NO	
3.2) Is your software available for free?	YES	YES	YES	
3.3) Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	YES	YES	YES	
3.4) Is your software hosted in an established, third-party repository like GitHub (https://github.com), BitBucket (https://bitbucket.org), LaunchPad (https://launchpad.net) or SourceForge (https://sourceforge.net)?	YES	YES	YES	
Q4 - Your software's documentation				
4.1) Is your documentation clearly available on your website or within your software?	YES	YES	YES	
4.2) Does your documentation include a "quick start" guide, that provides a short overview of how to use your software with some basic examples of use?	YES	YES	YES	
4.3) If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?	YES	YES	YES	
4.4) Do you provide a comprehensive guide to all your software's commands, functions and options?	YES	YES	YES	
4.5) Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	NO	NO	NO	
4.6) If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	NO	NO	NO	
4.7) Do you store your documentation under revision control with your source code?	N/A	N/A	N/A	
4.8) Do you publish your release history e.g. release data, version numbers, key features of each release etc. on your web site or in your documentation?	YES	YES	YES	
Q5 - How you support your software				
5.1) Does your software describe how a user can get help with using your software?	YES	YES	YES	
5.2) Does your website and documentation describe what support, if any, you provide to users and developers?	YES	YES	YES	
5.3) Does your project have an e-mail address or forum that is solely for supporting users?	YES	YES	YES	
5.4) Are e-mails to your support e-mail address received by more than one person?	NO	NO	NO	
5.5) Does your project have a ticketing system to manage bug reports and feature requests?	NO	NO	NO	
5.6) Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	N/A	N/A	N/A	
Q6 - Your software's maintainability				
6.1) Is your software's architecture and design modular?	YES	YES	YES	
6.2) Does your software use an accepted coding standard or convention?	YES	YES	YES	
Q7 - Open standards and your software				
7.1) Does your software allow data to be imported and exported using open data formats? e.g. GIF, SVG, HTML, XML, tar, zip, CSV, JSON, NetCDF, or domain specific ones	YES	YES	YES	
7.2) Does your software allow communications using open communications protocols? e.g. HTTP, FTP, XMPP, SOAP over HTTP, or domain-specific ones	YES	YES	YES	
Q8 - Your software's portability				
8.1) Is your software cross-platform compatible?* e.g. does it run under two or more of Windows, Unix/Linux and Mac OS X, or can be used from within two or more of Internet Explorer, Chrome, Firefox and Safari?	YES	YES	YES	
Q9 - Your software and accessibility				
9.1) Does your software adhere to appropriate accessibility conventions or standards?	YES	YES	YES	
9.2) Does your documentation adhere to appropriate accessibility conventions or standards?	YES	YES	YES	
Q10 - How you manage your source code				
10.1) Is your source code stored in a repository under revision control?	YES	YES	YES	
10.2) Is each source code release a snapshot of the repository?	YES	YES	YES	
10.3) Are releases tagged in the repository?	YES	YES	YES	
10.4) Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	YES	YES	YES	
10.5) Do you back-up your repository?	NO	NO	NO	
Q11 - Building and installing your software				
11.1) Do you provide publicly-available instructions for building your software from the source code?	YES	YES	YES	
11.2) Can you build, or package, your software using an automated tool?* e.g. Make (https://www.gnu.org/software/make/), ANT (http://ant.apache.org/), Maven (https://maven.apache.org/), CMake (https://cmake.org/), Python setuptools (https://pypi.python.org/pypi/setuptools), or R package tools (https://cran.r-project.org/doc/manuals/r-devel/R-exts.html)	NO	NO	NO	
11.3) Do you provide publicly-available instructions for deploying your software?	YES	YES	YES	
11.4) Does your documentation list all third-party dependencies?	YES	YES	YES	
11.5) Does your documentation list the version number for all third-party dependencies?	YES	YES	YES	
11.6) Does your software list the web address, and licences for all third-party dependencies and say whether the dependencies are mandatory or optional?	NO	NO	NO	
11.7) Can you download dependencies using a dependency management tool or package manager?* e.g. Ivy (http://ant.apache.org/ivy/), Maven (https://maven.apache.org/), Python pip (https://pypi.python.org/pypi/pip) or setuptools (https://pypi.python.org/pypi/setuptools), PHP Composer (https://getcomposer.org/), Ruby gems (https://rubygems.org), or R PackRat (https://rstudio.github.io/packrat/)	YES	YES	YES	
11.8) Do you have tests that can be run after your software has been built or deployed to show whether the build or deployment has been successful?	YES	YES	YES	
Q12 - How you test your software				
12.1) Do you have an automated test suite for your software?	YES	YES	YES	
12.2) Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	NO	NO	NO	
12.3) Do you use continuous integration, automatically running tests whenever changes are made to your source code?	YES	YES	YES	
12.4) Are your test results publicly visible?	YES	YES	YES	
12.5) Are all manually-run tests documented?	YES	YES	YES	
Q13 - How you engage with your community				
13.1) Does your project have resources (e.g. blog, Twitter, RSS feed, Facebook page, wiki, mailing list) that are regularly updated with information about your software? e.g. release announcements, publications, workshops, conference presentations	NO	NO	NO	
13.2) Does your website state how many projects and users are associated with your project?	NO	NO	NO	
13.3) Do you provide success stories on your website?	NO	NO	NO	
13.4) Do you list your important partners and collaborators on your website?	N/A	N/A	N/A	
13.5) Do you list your project's publications on your website or link to a resource where these are available?	NO	NO	NO	
13.6) Do you list third-party publications that refer to your software on your website or link to a resource where these are available?	NO	NO	NO	
13.7) Can users subscribe to notifications to changes to your source code repository?	YES	YES	YES	
13.8) If your software is developed as an open source project (and not just a project developing open source software), do you have a governance model?	NO	NO	NO	
Q14 - How you manage contributions				
14.1) Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?	YES	YES	YES	
14.2) Do you have a contributions policy?	YES	YES	YES	
14.3) Is your contributions policy publicly available?	YES	YES	YES	
14.4) Do contributors keep the copyright/IP of their contributions?	YES	YES	YES	
Q15 - Your software's copyright and licensing				
15.1) Does your website and documentation clearly state the copyright owners of your software and documentation?	YES	YES	YES	
15.2) Does each of your source code files include a copyright statement?	NO	NO	NO	
15.3) Does your website and documentation clearly state the licence of your software?	YES	YES	YES	
15.4) Is your software released under an open source licence?	YES	YES	YES	
15.5) Is your software released under an OSI-approved open-source licence?	YES	YES	YES	
15.6) Does each of your source code files include a licence header?	NO	NO	NO	
15.7) Do you have a recommended citation for your software?	YES	YES	YES	
Q16 - Your plans for the future				
16.1) Does your website or documentation include a project roadmap (a list of project and development milestones for the next 3, 6 and 12 months)?	YES	YES	YES	
16.2) Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	NO	NO	NO	
16.3) Do you make timely announcements of the deprecation of components, APIs, etc.?	NO	NO	NO	