Borealis Web App☆

A.A. Anisimov[a,*]

[a]*Peter the Great St.Petersburg Polytechnic University*
[b]*Institute of Computer Science and Technology*

Abstract

This paper is about developing front-end for Borealis. Borealis is a bounded model checking tool for C programming language.

Keywords: bounded model checking, Borealis, Django, Python

1. Introduction

This work is about creation web application for Borealis a bounded model checking system for C language. Due to the lack of early developments in this area, there are no requirements for the technologies and tools used. By results of the analysis of approaches, the Python language and the Django web framework were chosen to implement the application. Because verification by Borealis can take a considerable amount of time, they should be performed asynchronously with subsequent notification of the user about the results. For work with asynchronous tasks using the system of asynchronous performing Celery tasks. Issuance of the result in the form of source code with highlighted defects is done with the help of the library Pygments. Since it is assumed that the user can test entire projects, there is a need to access user repositories on GitHub and BitBucket sites. For authentication and access to user repositories, the social-app-django library is used. To store user history, the SQLite3 database is used, which is accessed through Django ORM. Borealis itself is supplied in the form of a Docker image.

2. Architecture

Since we selected Django as the implementation tool, the implementation will be subordinated template MVC, the general scheme of which is represented in the figure 1

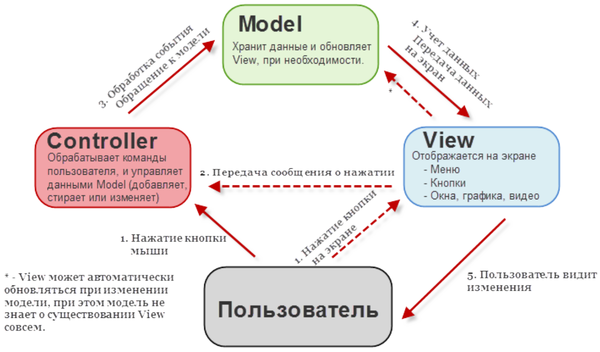
Model
Хранит данные и обновляет View, при необходимости.
Controller
Обрабатывает команды пользователя, и управляет данными Model (добавляет, стирает или изменяет)
View
Отображается на экране
- Меню
- Кнопки
- Окна, графика, видео
Пользователь
1. Нажатие кнопки мыши
1. Нажатие кнопки на экране
2. Передача сообщения о нажатии
3. Обработка события Обращение к модели
4. Учет данных Передача данных на экран
*
5. Пользователь видит изменения
* - View может автоматически обновляться при изменении модели, при этом модель не знает о существовании View совсем.

Figure 1: MVC general scheme

Following the template, it remains to bring the UML flowcharts of the Model.

And doesn't have any central architecture, because it's main idea to support model and template design.

[*]Master student, group 13541/3.
Email address: `pilotofsparrow@gmail.com` (A.A. Anisimov)

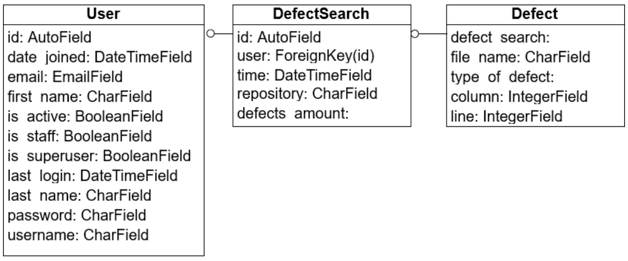

Figure 2: UML diagramm for model

3. Development

Thanks to the concept of rapid development Django, serious difficulties in the development process was not met. The most difficult was to introduce a system for highlighting defects. For its development, the Pygment library was used. As a result of the development, all the goals were achieved.

4. Conclusion

As a result of the work done, all the tasks were achieved and a usable web application was obtained. The user is provided with a convenient web interface for the Borealis limited model verification system. For those who want to try Borealis on small projects the main page provides an intuitive friendly interface and an editor with backlight C code. For larger projects in need of verification, there is a possibility checking entire repositories, you need to authenticate via BitBucket or GitHub, with subsequent access to the list of user's repositories. After the end of the test, the user is e-mail comes the notification of the number of defects. After logging into the history, the user will be able to view in an accessible format all the errors found in the sent to check code.