Project Jabberwocky

Functional Specification

Lucas Chaufournier

2014

# Overview

Project Jabberwocky is an evolving integrated development environment that grows with a programmer as they learn to program. Too often new programming students learn to code in big clunky environments with features that are designed for experts. These features such as auto completion and auto-generation of classes, serve to act as a crutch for new programmers as they rely on them too much rather than learning on their own. Project Jabberwocky starts out as simple text editor with only syntax highlighting enabled. As a student writes programs in the editor the editor learns more about what the programmer knows and will proceed to unlock more features as it sees fit. By the time students have advanced all the levels, students will have features such as copy and paste, syntax completion, auto-generation of classes, a debugger and more.

# Scenarios

**Scenario 1: Alice**

Alice is an ambitious student who wants to learn programming on her own. She finds programs such as eclipse and vim too confusing to use and wants a simple system that will assist her rather than hamper attempts at learning programming. She decides to give Jabberwocky a try so she downloads the editor and starts it up. She is presented with a simple web interface that is just a text editor. She starts programming on it but because she is struggling on her own she soon gets stuck on the difficult task of implementing a list. The system realizes that she is struggling and offers to recommend a resource she can use to learn more about how to use a list. She visits the reference and is soon on her way.

**Scenario 2: Bob**

Bob is a student Professor Williams’ class. Professor Williams has instructed his students to use Jabberwocky as their coding editor so that he can tell how well his students are learning to program. Bob logs onto the class website and enters his provided account information. He is presented with a level one editor that only allows him to enter text. Bob works on his first assignment and soon completes it with ease. As the semester progresses on Bob continues to work on his assignments and after the fifth assignment, the editor announces that he has unlocked a new level. The editor now has the functionality that he can use auto-completion and auto-generation of classes. Throughout the rest of the semester as he unlocks assignments, he is able to unlock more levels and at the end of the course he has a full featured editor that he can use for the rest of his programming classes.

**Scenario 3: Professor Williams**

Professor Williams is an instructor at a local university. Because he is a new professor, he is assigned to teach the intro programming class for freshmen. Professor Williams decides to use Jabberwocky to assist in his class so he can monitor the progress of his students. Professor Williams logs onto the website and creates a new classroom environment. He is presented with a control panel that allows him to see the roster of his class, the level that their editor is at and a control panel. At the control panel Professor Williams can adjust the rate at which features are unlocked, as well as the specific features that can be unlocked. He can also unlock different features for different students. This allows him to have a fully configurable environment for his students.

# Non Goals

For the initial release of Project Jabberwocky there will be no support for multiple classes. Students will need to have a separate account for each class. Teachers will also need to have an account for configuring the classroom and one for using the actual editor.

# Initial Login Page

When a user visits the Project Jabberwocky page they are presented on the left with a brief overview of what the project is and some of its features. On the right hand side is a login box with ability to login, create an account, or to create a class. The login box takes in an email as well as a password. When a user clicks the submit button their account information goes against several checks. The system first validates the information is a valid input. If the email is not recognized then an error is displayed alerting the user to try again or to create an account. If the email is correct, then system checks their account email against the database and if it is found, the password is checked. If the password matches the stored value, they are directed to the editor window if they are a student and to the instructor panel if they are an instructor. If the password is incorrect, the user is alerted and they are instructed to try again. If they get it wrong too many times they are locked out for 5 minutes. If a user clicks on the account creation button they are directed to the Account Creation Page. When a user clicks on the create class button they are directed to the Class Creation Page.

# Account Creation Page

When the user clicks on the account creation button, they are presented with this page. The account creation page presents a form that asks for several pieces of identifying information. The user is asked to present a valid email, a valid password, a first and last name, and a course identification number. When the user clicks the submit button the data is then validated. The email is checked against the system to see if it matches and existing entry. If the email is found in the database, the user is alerted and told to use a different email. The password is checked against its second entry to make sure they match and have not been mistyped. The course identification number is checked to make sure it matches an existing class. If the course number is not valid the user is then alerted to check and renter the value.

Once the data has been successfully validated, it is then entered into the database as a user entry. The user is then added to the class and sent an email confirming their account creation. The email provides a confirmation message that presents the user name of the user as well as the name of the course they have enrolled in. The user’s password is not contained in the email. The system then directs the user to the Editor Window so the user can start working away.

# Class Creation Page

Once a user has clicked on the create class button they are presented with this page. The class creation page presents a form that allows an instructor to enter account details and to fill in information about their class. The form contains the following information: First name, Last name, Email, Password, Course name and Expected number of students. When the instructor hits submit the data is validated. The instructors email is checked against the system to see if the email is already taken. If the email is already present the user is alerted to choose a different email. The two passwords are checked against each other to make sure that the user typed their desired password in correctly. If they do not match the user is instructed to please type in their password again.

When the data has been successfully validated, the account information is saved as a user record in the database and the course information is added as a course record. The system then sends a confirmation email to the instructor. The email contains a confirmation message as well as the username and class information including a generated course code to be used for student registration. The system then directs the user to the Instructor Control Panel.

# Editor Window

Once a student has successfully logged in, they are presented with the editor window. It is at this window that the student will spend all their time. As a level one editor the student is allowed to type text and have syntax highlighting, all other features including copy/paste are restricted. Once a student has completed an assignment they will hit the build button. The build button will scan over their code and check it for correctness. If the code is deemed correct the editor will keep a record and then the build process will continue. If a student’s code is deemed incorrect the editor will keep a different record and then the build process will continue.

Once a student has completed enough programs the editor will examine the correct vs incorrect indicators and determine if the student can move on to the next level. If the editor determines the student can move onto the next level, the student is alerted with a pop up detailing the new features. The level will be recorded in the students profile record. This indicators are reset back to zero and the process repeats for the new level.

If a student builds a program enough times with the same error, the editor will attempt to identify the error and provide relevant resources to assist the student in learning the topic. This will not be possible for all topics but will function for basics like looping, conditionals, stacks, arrays, etc. These relevant resources will be either to detailed visuals of data structures or to online explanations of the topic.

If a student has advanced all the levels they will be presented with a congratulatory message and be able to use the editor as full editor for future projects if they so choose.

# Instructor Control Panel

Once an instructor has logged in or created a course they are directed to the Instructor Control Panel. The control panel has a left hand side bar with navigation for the roster as well as the classroom settings. On the right is the current context the instructor is selected on, the default is the roster. At the top of the page the course registration number is displayed in large text at the top of the window, and will not change.

In the roster context, the instructor is presented with a large table that shows the students information along with the current level of their editor and a features button. The features button allows the instructor to unlock certain features for a user. The roster also allows a professor to downgrade or upgrade a student’s level at will. When an instructor makes changes to a student, the changes are sent to the database where they are stored in the students profile record. Students individual setting override the overall classroom settings.

In the classroom settings context, the instructor is presented with configuration options for the classroom. Here the instructor can unlock or even lock features for the entire classroom as they see fit. The instructor can also specify the leveling pace for the classroom, so if they desire functionality to be unlocked slower they can turn the difficulty up or they can turn it down to make leveling faster. As the professor make overall changes they are stored in the classroom profile record.