Product Backlog

Team 5

Daniel Cruz, Himal King, Nick Schumacher, Josh Stuckey, Cody Tyson, Michelle Young

Problem Statement

With the thousands of classes available on Purdue University’s campus, it is sometimes difficult to decide which ones to take and who to take them with. Our solution is to create a centralized place for students to anonymously write a review for any class available. Our product will be designed for Purdue students to make assessments on what courses to take based off of ratings from other students.

Background Information

Students often ask their friends who have already taken the classes how difficult it was, what the professor is like, and if they recommend taking it. However, it is impossible to have a friend who has taken every class on campus. Ratemyprofessor.com is an option for reviewing professors, but it wasn’t designed for reviewing courses. While taking a class, students are provided the option, through the university, to write a review for their course, but it is strictly for the professor’s benefit; the reviews are not available to be seen by other students. There are some websites, like www.locazu.com, that offer a way to review a course you have taken, but these sites are not Purdue specific and it can be difficult to locate relevant reviews.

Environment

The development environment is simply a machine with Ruby on Rails installed. Rails can be installed on Windows, Mac, or Linux OS’s. The languages included in this project are Ruby on Rails, HTML5, Javascript, JQuery, and CSS. The database we use is SQlite in development and PostgreSQL on our live website.. Bootstrap will be used to simplify the styling of the web pages.

Requirements

Functional

|  |  |  |
| --- | --- | --- |
| Backlog  ID | Functional Requirements | Hours |
| 1 | As a user, I would like to publish a review for any course I have taken at Purdue University | 8 |
| 2 | As a user, I would like to be able to search for courses I have taken at Purdue University or courses I am interested in taking | 16 |
| 3 | As a user, I would like to rate my class based on whatever aspects I feel (difficulty, workload, fairness of grade received, etc…) | 8 |
| 4 | As a user, I would like to be able to log in with my purdue email and password as credentials | 8 |
| 5 | As a user, I would like to be able to pick my professor(s) when I give a review on a class | 6 |
| 6 | As a user, I would like to be able to see the highest rated courses in the department I choose | 6 |
| 7 | As a user, I would like to be able to recover my login if I forget my password | 4 |
| 8 | As a user, I would like to be able to flag reviews as inappropriate content | 4 |
| 9 | As a user, I would like to see when a review was published alongside the review (timestamp) | 2 |
| 10 | As a user, I would like to know the average grade that students received in the class | 4 |
| 11 | As a user, I would like to upload and see uploaded syllabi for the class | 4 |
| 12 | As a user, I would like to add the names of TA’s in my review | 2 |
| 13 | As a user, I would like to help decide if a review is helpful or not | 4 |
| 14 | As a user, I would like to be able to see the top rated courses on the home page | 6 |
| 15 | As a user, I would like to be able to ask questions about a class and have other users answer them | 6 |
| 16 | As a user, I would like to give an estimated amount of hours spent each week on the class work outside of class | 2 |
| 17 | As a user, I would like to be able to see whether the textbook for the class was required. | 2 |
| 18 | As a user, I would like to see a student provided course description that I or anyone else can edit. | 6 |
| 19 | As a user, I would like to provide feedback to the development team about their website | 4 |
| 20 | As a user, I would like to upvote or downvote the answers to questions | 4 |
| 21 | As an admin, I would like a page to see all the flagged posts so I can decide what to do when them. | 4 |

Non-Functional

|  |  |
| --- | --- |
| Backlog ID | Non-Functional Requirements |
| 1 | Once a review has been flagged, it is hidden from the list of reviews that are displayed |
| 2 | The web application should have a modern style to it |
| 3 | The web application should be intuitive |
| 4 | The web application should be compatible with any screen size. |
| 5 | Logins and passwords should be encrypted. |
| 6 | When our website is being hosted it should be it should come back up if the servers go down. |
| 7 | We should have a backup of our databases just in case the databases are lost |

Use Cases

|  |  |
| --- | --- |
| **Case: Write a review** | **System Responses** |
| 1. Type the course name/number into the search bar | 2. suggestions appear as the user types into the searchbar |
| 3. click submit | 4. search results are loaded sorted by rating |
| 5. click on the course to write a review | 6. the course page is opened |
| 7. type the review, give a rating, fill in other fields (professor name, TA names, hours spent on work, upload syllabus, avg. grade in class, etc…) |  |
| 8. click submit | 9. the review and rating is published to the course’s page and timestamped |

|  |  |
| --- | --- |
| **Case: Vote on answers to questions** | **System Responses** |
| 1. log into the site | 2. user preferences are loaded |
| 3. navigate to a course page |  |
| 4. click on the helpful or not helpful buttons | 5. a counter is incremented for that review |

|  |  |
| --- | --- |
| **Case: Sign up** | **System Responses** |
| 1. Click sign up button | 2. Sign up webpage loads |
| 3. Type in email and password of choice |  |
| 4. click submit button | 5. Check to make sure email is not a duplicate, and it is also a purdue.edu email |
|  | 6. If #5 passes, send user to home page. Else, send user back to signup page with appropriate error. |

|  |  |
| --- | --- |
| **Case: Log In** | **System Responses** |
| 1. Click login button | 2. Login webpage loads |
| 3. If the password is forgotten, click the password recovery button | 4. Load password recovery page. |
| 5. Type in email to recover password |  |
| 6. Click submit | 7. Send email to user with a login link |
| 8. If password is not forgotten, type email and password | 9. Check to make sure the password and email match |
|  | 10. 6. If #9 passes, send user to home page. Else, send user back to login page with appropriate error. |

|  |  |
| --- | --- |
| **Case: Provide feedback for reviews** | **System Responses** |
| 1. log into the admin account | 2. admin privileges are unlocked |
| 3. navigate to the reviews page | 4. the review page where flagged reviews are posted is loaded |
| 5. the admin can confirm or disregard flagged reviews | 6. confirmed reviews are deleted and disregarded reviews have their flagged count reset to 0 |

|  |  |
| --- | --- |
| **Case: Provide feedback for reviews** | **System Responses** |
| 1. Navigate to a course page |  |
| 2. Tap “yes” or “no” as to whether or not a review is helpful | 3. An alert is flashed thanking the user for the feedback |

|  |  |
| --- | --- |
| **Case: Flag review as inappropriate** | **System Responses** |
| 1. Navigate to a course page |  |
| 2. Tap on the flag next to a review | 3. the review is hidden and sent to site administrators for review |

|  |  |
| --- | --- |
| **Case: Ask questions about a course** | **System Responses** |
| 1. Navigate to course page |  |
| 2. Type into “new question” field a new question |  |
| 3. click submit | 4. The question is posted to the page and a reply field is generated below the question |

|  |  |
| --- | --- |
| **Case: Respond to questions about a course** | **System Responses** |
| 1. Navigate to course page |  |
| 2. Click in the reply field below a question and type out a response |  |
| 3. Click submit | 4. the reply is posted to the page under the question and another reply field is generated below the question |

|  |  |
| --- | --- |
| **Case: Set homepage preferences** | **System Responses** |
| 1. after logging in and reaching the homepage, click on the “Preferences” button at the top | 2. a dialog box opens prompting the user to choose their major |
| 3. click submit to save changes | 4. The preferences are saved for that user. The homepage is reloaded with their major’s courses listed |

|  |  |
| --- | --- |
| **Case: Provide feedback to site admins** | **System Responses** |
| 1. Click on the “contact us” button | 2. The contact page is loaded |
| 3. Fill out the email, subject, body fields |  |
| 4. click send | 5. an email is sent to the administrator account |

|  |  |
| --- | --- |
| **Case: Edit a course description** | **System Responses** |
| 1. log into the website | 2. user preferences are loaded |
| 3. navigate to a course page |  |
| 4. click on the course description field to edit |  |
| 5. click on the submit button when finished editing the description | 6. the new description is loaded as the courses description |