**Vision Statement**

What makes Spot-On different is that, unlike Google search, which focus on multiple search inquiries, Spot-On only works on search and crawling through deep search engines to bring customers the top products with best rates and reviews. Spot-On does nothing else but that and because of it single focus, it performs very good**!**

**Project Charter**

|  |  |
| --- | --- |
| **Project Title:** Spot-On | |
| **Start Date:** 01/29/2018 | **End Date:** Until the end of the class |
| **Project Manager:** Martin Mailloux  **Project Teams:** Jonathan Diaz, Lam Nguyen, John Asare | |
| **Project Sponsor:** No one yet | |
| **Customer:** Online Shopping People Read Reviews Before Shopping | |
| **Users:** Of all ages, non-technical and technical | |
| **Purpose (Problem or opportunity addressed by the project)**  Most users rely on Google, Bing and other search engines as their source of searching for reviews on products. The way that these search engines serves customer is by crawling through the internet and returning reference of other websites that will again requires the customers and the users to again explore through the suggested site for product’s reviews. This behavior has become a norm and users might be getting used to it but a better way of design can be born.  The purpose of this projects is to produce a friendly graphic user interface that most users are familiar with but to only shift the software focus on crawling through the internet to results a product with a best reviews and rates. As users come to visit the site, their expectation wouldn’t be a references of websites for them to explore through in order to get a product with best reviews but their expectation will be getting the actual product. | |
| **Goals and Objectives**  The general goal of the project is to enhance the old traditional experience of searching for products on the internet. More specifically:   * The website/application will be user friendly. It won’t require any other additional learning curve. It will keep the old traditional graphic user interface but much cleaner. * The application or the website should be able to give the user’s need in a quick and user friendly mannner * The application should at least deliver two results:  1. The product that the customer is looking for 2. The reviews and the option to buy when they product is up | |
| **Deliverables Planning Calendar**  The following milestones are planned. The dates are very rough estimates. They should not be made public outside of the immediate project team. Rough estimate for project duration is until the class is over. The schedule below is work that need to be done in class on every class period.  02/10/2018 - Picking the project materials and software 02/14/2018 - Project proposal in class and presentation  03/01/2018 - Getting ready to demo the database and the web spider 03/03/2018 - Assign issues on Jira being completed 03/07/2018 - Team learning Intro to Django  04/05/2018 - Technical Risks Resolved (Deliverable: technical prototype that demonstrates programming elements needed to implement desired functionality) 04/12/2018(week 3) – Database and SQL implementation complete  04/26/2018(week 2) - Architecture design and software documentation, HTML and CSS deployment for the project 04/20/2018 – Fixing bugs, finishing off CSS, fix database, documenting and preparing for final presentation  05/13/2018 (week 2) – Deadline and final demo | |
| **Project Management**  **Software Practice**  We will be using an ***Agile Software Development***. Specifically, the ***Adoptive Software Development*** method. Iterative and incremental approach will be issued on Jira with deadlines. As a software is ready, we will push it to our internal software and test it. No iteration will favor technical “infrastructure” over usable functionality.  Since we are learning a new programming language, the Agile deployment is being delayed. Because of this, teams have to do extra work on their own to time. Team members will be taking extra online courses about Django. However, regardless of the extra time we all put in on learning Django, every team is responsible for meeting submitting their work. Whether is it done or not.  **Issues and Task (Jira and GitHub)**  We use Jira and GitHub to issues tasks for each team. A team member can assign an issue for his self if needed. Also, when a team a member sees a bug in the project, that team member issues a task for the responsible team member who is in charge of the code. We use both GitHub task and Jira interchangeably.  **Communication**  Besides using Jira and GitHub as a sort of communication, we also have a Facebook group chat as way to discuss a short idea.  Sometimes we use GitHub’s comment as also a sort of communication but the most primary way to get in contact with the team is Facebook group chat. | |
| **Constraints**  We will be using Django and other database server. Doing so, we will be relying on those third parties. Anything they do (such as updates and bugs) will also affect this project. This maximize our dependencies.  Our team seems not to have a very strong with the Django and SQL language. We are however learning from basics and applying it to the project at the same time. Therefore, we are doing two things as one and it slowing down our deliverables. In doing so, it delayed some of our project’s deadlines and slows down the projects. | |
| **Quality Assurance**  We are not using any hosted server that hosts our project but, we are using Django local server. With that, team members can individually run the server on their local server on port 8080. After running it, a team member will test every function and if sees a bug or error, he will report it on Jira or GitHub.  Also, we meet once a week to run the project. Each member will use it and if any error occurs, we both work on it. Also we use GitHub history code tracking system to go back to our project and fix bugs.  On top of that, we also run the server for our friends to use it and gives us feedback. They give feedback depending of the interface, functionality, design and speed flow. | |

**Revision Control**

You can also locate it under [Spot-On Project](https://github.com/nctl144/spot-on) on GitHub.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Modified By** | **Date** | **Description** |
| 1.0  (Created GitHub) | Lam | 02/22/2018 | Created the GitHub repository. Teams fork the original branch and then tested the connection. We all push our first commit as a double test |
| 1.0.1  (Created GitHub  branch) | Lam, John &  Jon | 02/25/2018 | Each team has created a branch and is already pushing their project to their branch. We made a pull request for the master branch to review the code then add it to the internal code |
| 1.0.2  (Demo project) | Lam, John &  Jon | 03/02/2018 | We have created a demo project for us to refresh our brain and to learn a new coding software. We have decided to go with Django. Now, we are going to be working on a demo project to master the basics of Django |
| 1.0.3  (Documentation) | John | 02/27/2018 | I created the software and the architecture documentation for submission. The documentation have both a summary of the project and class diagrams |
| 1.1.2 | Lam | 03/01/2018 | IDLE. Lam decided that we should all use Atom as our main idle environment but it didn’t work out so team members has to shift all their code to their prefer idle and start pushing it on GitHub |
| 1.1.3  (Scrappy and Data) | Jon & Lam | 03/22/2018 | They both decided on the database (SQL). They then added database to the SQL table. The database was then connected to the Django environment |
| 2.0  (Project Spot-On) | Lam | 04/01/2018 | The demo-project was created and moved into a new repository. All the code from demo-project was moved and talking to each other. Django server was running the project on local port 8080. The project was demo in class today. It access the database and return a results of shoes so far |

|  |  |  |  |
| --- | --- | --- | --- |
| 2.0  (Spot-On) | Lam & Jon | 04/01/2018 | Added a static folder |
| 2.0  (Spot-On) | Lam & Jon | 04/07/2018 | Added a scapy-spider which crawls data from the Nike server. Tested the Json |
| 2.0  (Spot-On) | John | 04/17/2018 | Created a new html view page as an introduction for the project |
| 2.1 | Jon | 04/19/2018 | Added details of the shoes to the database for best searching |