Software Engineering Document

Review Aggregation Website

Team 2

Riyo Aloshyas
Hyeoncheol Kim
Md Siamul Islam
Cody Smith
Muhammad Jamal

Table of Contents

Introduction

There are a lot of people who purchase items through E-commerce websites. By online shopping, many E-commerce companies were not able to find whether the customers are satisfied with the services provided by the firm. For this reason, we develop a system where various customers give reviews about the product and online shopping services, which in turn help the E-commerce enterprises and manufacturers to get customer opinion to improve service and merchandise through collecting customer reviews. In this system, an algorithm will be used to track and manage customer reviews through collecting topics and opinions from online customer reviews. Our user will be able to view many products and get a link to purchase the product. User can see categorized reviews by searching for certain keywords. The keywords which are already exist in the database will be matched based on the comparison. The system will rate the product and services provided by the enterprise.

Business Landscape

The idea of the website is to compile reviews from different platforms online for any product being searched by any user. This will allow users to post a review on a product that can build value in a product and make the user more inclined in buying the product based on the reviews. The website will work as a key marketer for any new product being launched. The benefit of the website would be that it will contain information from several platforms and will contain reviews from multiple websites and the user will be able to find the most relevant review based on the search. This can also be a great way for users to share their particular experience with any new product which can be beneficial to the client and help grow more business for the product. In today’s world reviews are viewed very importantly by any individual spending money.

Requirements

The following lists the core functional capabilities of the Review Aggregation Website:

Functional Specifications:

Product Management System:

1. View product reviews
2. Search feature
3. Create product list
4. Add/delete products
5. Modify existing products
6. Organize products by category

Review Management System:

7. Return results based on user input
8. Select from various review categories
9. Save favorite products
10. Start initial search and keep pulling remaining entries
11. Allow users to post their own reviews on products
    - 11.1. Include ReCAPTCHA Verification

User Interface:

12. Framework for Website
13. Show dialogue box for user to type in and initiate search
14. Show navigation bar
15. Create and manage user and admin accounts
16. Select from various review categories
17. Save favorite reviews
18. Refresh
19. Manage ads
20. Track users
21. Manage preferences
22. Display error message dialogue box if no results found
23. Display support contact information

Data Storage System:

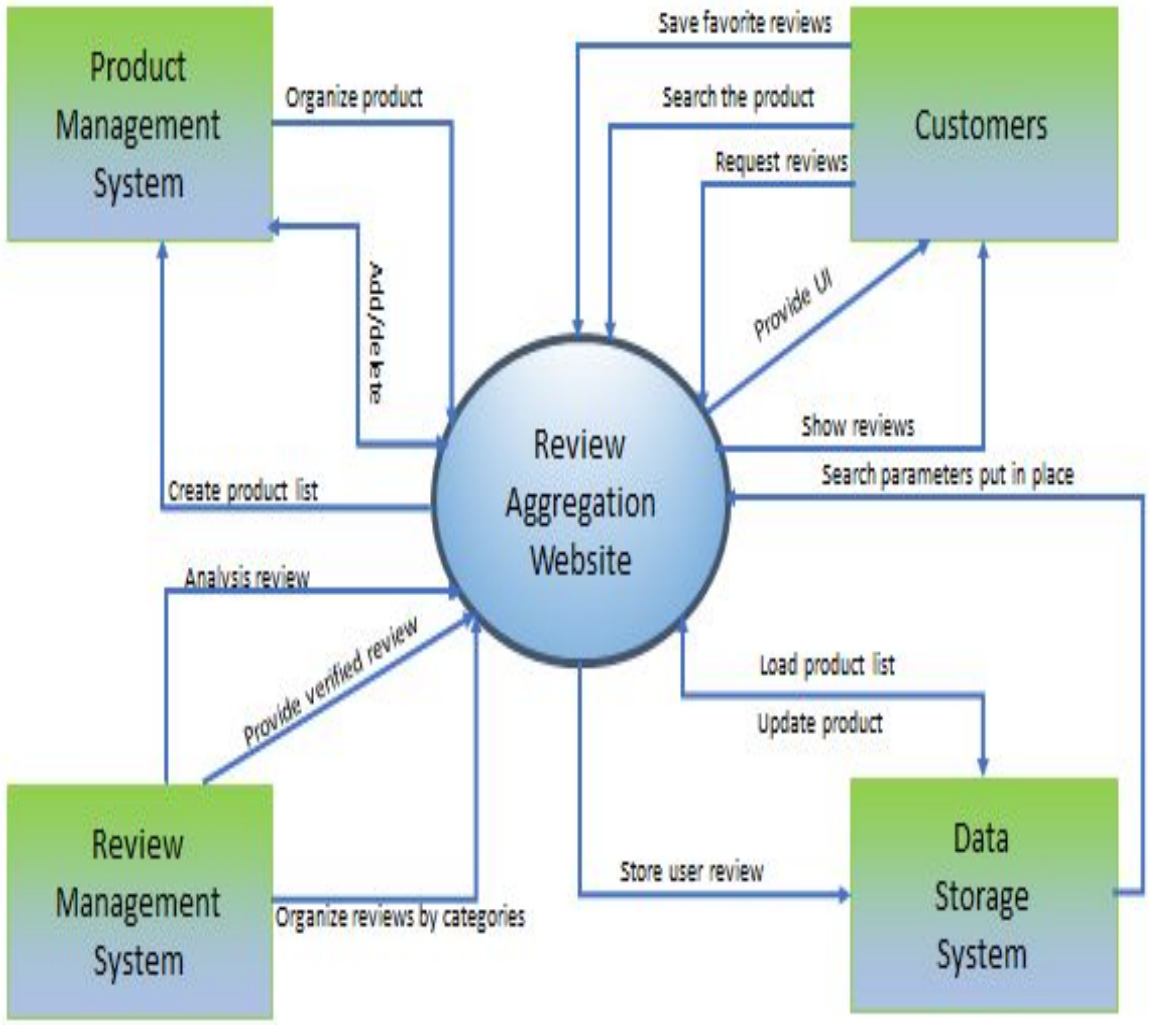
24. Load product list
25. Update
26. Search parameters put in place
27. Store user reviews

Non-Functional Specifications:

28. View 100 reviews per page
29. Display review results within .5 seconds and continue to build more
30. Limit searches to 50 words to provide the most accurate results
31. Save favorite reviews within .5 seconds

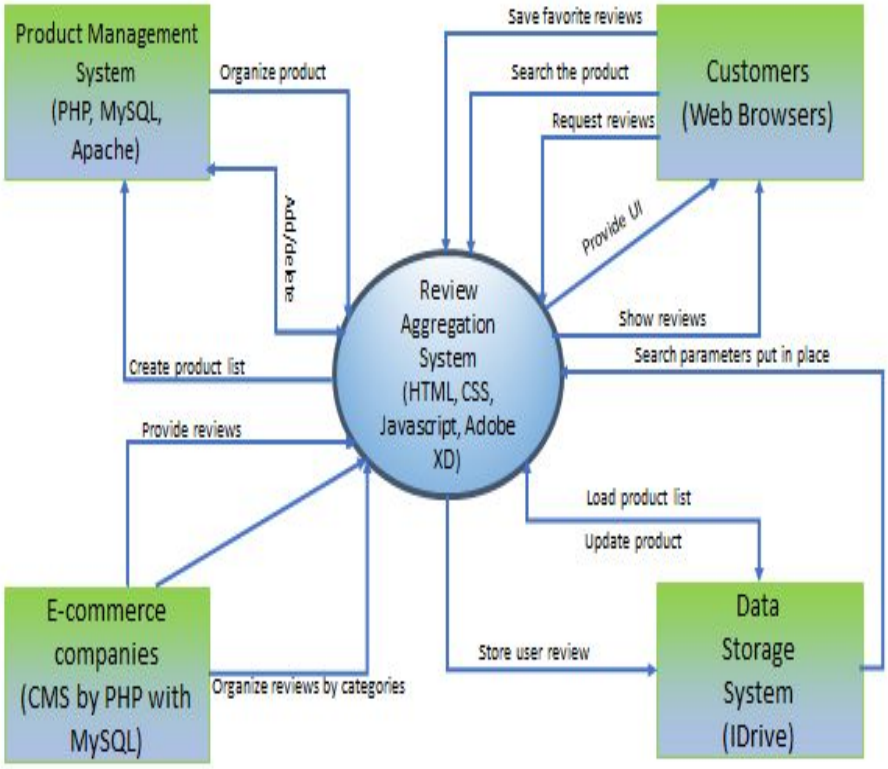
32. Display category list within .5 seconds when user hovers over “Category”
33. Add/delete reviews within .5 seconds
34. Log in user within 2 seconds
35. Create accounts within 2 seconds after user provides necessary information
36. Store and display user reviews within .5 seconds
37. Limit ads to a maximum of 4 per webpage
38. Display error messages within 1 second
39. Display a refreshed web page within 2 seconds
40. Display search results using preferences within .5 seconds

Architecture/Design



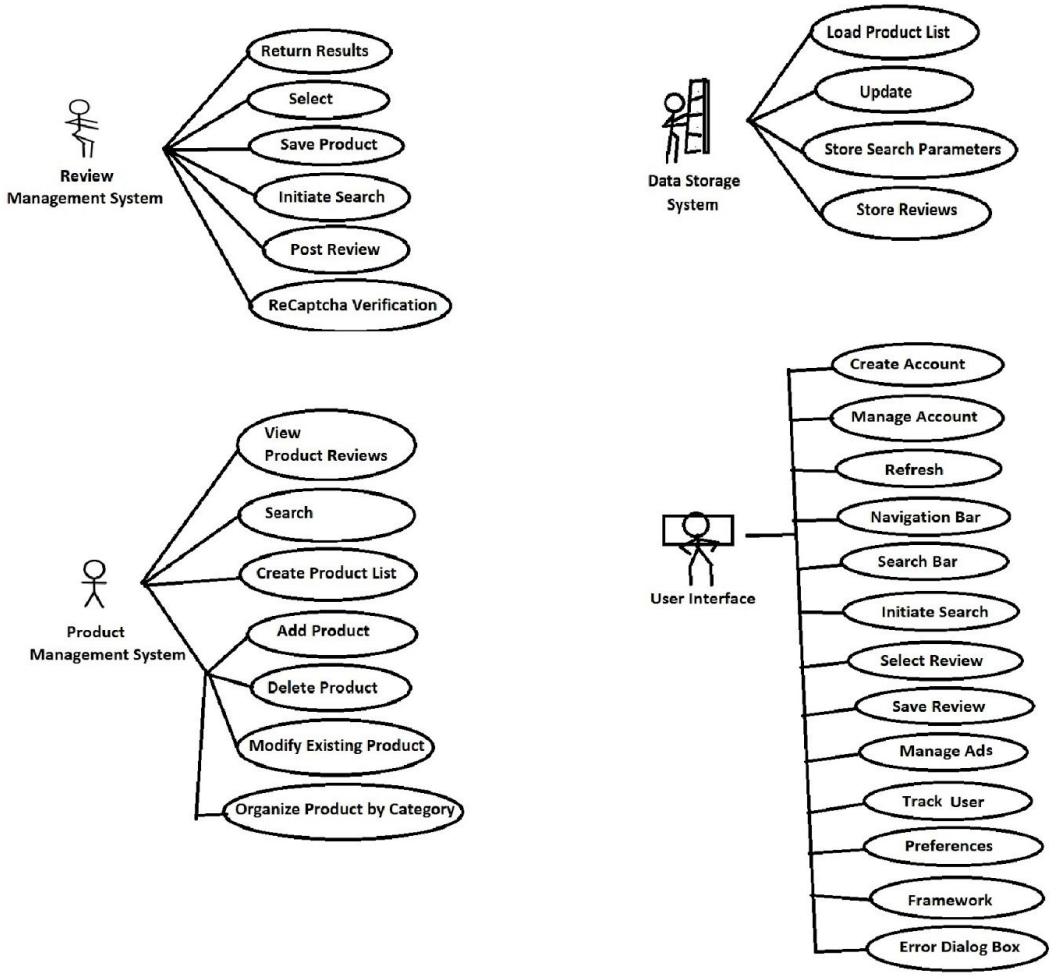
The context diagram above illustrates the relationship between the four components and the overall system, showing how each component satisfies various requirements of the system.

Mid-Level Design



The mid-level design above displays which technologies and environments are to be used in each component of the system. Technologies and environments were chosen according to how appropriate and effective they would be for databases, management systems, data storage and web development.

USE CASES



The use cases illustrate how each component is related to various requirements of the system. Use cases are important because the test cases defined in the document may involve multiple system components and testing the use cases hence forces interaction between these systems.

Test Cases and RTM

Test Case Number	Req. #	Requirement	Execution	Acceptance Criteria
TC_001	1	Product Review Display Test	1. Products listed on the website will be tested to ensure all reviews pertaining to a product will be displayed to the user. 2. Products without reviews will be tested to display “No Reviews”. 3. Reviews will be tested to confirm at least 2 products from each product category (cell phones, tablets, home & office, etc…) will display one or more review(s).	1. For TC_001.1, all reviews for a specific product will be displayed to the user. 2. For TC_001.2, “No Reviews” will be displayed to the user when searching for a product without any reviews. 3. For TC_001.3, each product category must have a minimum of 2 products have one or more review(s) each.
TC_002	2	Search Feature Test	1. Search feature will be tested to ensure it accepts input. 2. Search feature will be tested to ensure search results return after entering input and pressing search button.	1. For TC_002.1, input must be accepted with a maximum of 32 words per search. 2. For TC_002.2, admin must be able to press search button after entering input. 3. For TC_002.2, search results should display based on search input. If no products

				match search input, then display “No Matches Found!”.
TC_003	3	Product List Creation Test	1. Product Management System (PMS) will be tested to ensure product lists can be created.	1. For TC_003.1, PSM should allow admin to create a product list.
TC_004	4	Product Addition/ Deletion Test	1. PMS will be tested to ensure products can be added to and deleted from a product list and the database.	1. For TC_004.1, products must be able to be added to a product list as well as the database. 2. For TC_004.1, products must be able to be deleted from a product list and the database.
TC_005	5	Existing Product Modification Test	1. Existing products will be tested to ensure that their specifications can be modified after being added to the product list and database.	1. For TC_005.1, the product list and database must be able to modify existing products up to date.
TC_006	6	Product Organization via Category Test	1. Products will be tested to ensure that they can be classified into specific categories.	1. For TC_006.1, admin must be able to select a category for a product to be placed in.
TC_007	7	Result Display after Search Feature Usage Test	1. Review Management System (RMS) will be tested to ensure that review results are returned after admin uses search feature.	1. For TC_007.1, input must be accepted with a maximum of 32 words per search and admin must be able to press search button after entering input. 2. For TC_007.1, search results should display based on search

				input. If no products match search input, then display “No Matches Found!”.
TC_008	8	Category Selection Test	1. RMS will be tested to select categories based on choice.	1. For TC_008.1, admin must have the ability to select categories.
TC_009	9	Favorite Product Save Feature Test	Uses TC_015 1. Using a user account, save a 25 products and retrieve saved items. 1. Attempt to save items without being logged in.	1. Acceptance if the saved list can be retrieved with 1 or more products saved. 2. Without being logged in, nothing should be found for saved list.
TC_010	10	Display Remaining Search Results Test	Uses TC_002, TC_028 1. Search for a product and receive an initial results list. 2. Ensure the next 100 products are still being retrieved. Once the “Show More” option is selected, show the 100 pre-gathered, and go get the next 100 products.	1. Passes if initial results are loaded. 2. Passes if next 100 results are loaded, and database retrieval is still gathering beyond the first 200 results.
TC_011	11	Allow users to post their own reviews on products	Uses TC_015 1. From no account, attempt to post a review. 2. From a user account, attempt to post a valid review.	1. Passes if error message displays when attempting to post review

			3. From a user account, attempt to post an invalid review.	without being logged in. 2. Passes if review from user account is successfully posted. 3. Passes if error message is displayed when invalid review is entered.
TX_011.1	11.1	Include reCAPTCHA Verification	Uses TC_011, TC_015 1. Attempt to post a valid review.	1. reCAPTCHA box must appear, and be passed in order to accept review and post it. Invalid reviews should not make it to this stage.
TC_012	12	Framework for Website	1. Open the frame on a display unit to ensure it matches the wireframe design, and elements appear where they should	1. Acceptance when website skeleton aligns with design plans
TC_013	13	Dialogue box for user to type in and initiate search	Uses TC_015 1. From no account and from user account view each webpage on website 2. Attempt to type a string into a search box	1. Passes if searchh bar is visible on every webpage, with or without a user account.

				2. Passes if search bar accepts strings to be entered.
TC_014	14	Navigation bar	Uses TC_015 1. From no account and from user account, use navigation bar on homepage to select different pages on website.	1. Passes if website successfully loads each page linked to on the navigation bar.
TC_015	15	Create and manage user and admin accounts	1. Attempt to create an admin account correctly. 2. Attempt to create an admin account incorrectly. 3. Attempt to create a user account correctly. 4. Attempt to make a user account incorrectly. 5. Login to admin account using wrong credentials. 6. Login to admin account using correct credentials. 7. Login to user account using wrong credentials. 8. Login to user account using correct credentials. 9. Use admin account to delete user account 10. Make changes to admin account 11. Delete admin account	1. TC_015.01 will pass if an admin account can be successfully created. 2. TC_015.02 will pass if system does not allow admin account to be created. 3. TC_015.03 passes if user account successfully created. 4. TC_015.04 passes if system prevents user account from being created. 5. TC_015.05 passes if login fails. 6. TC_015.06 passes if login is

				successful. 7. TC_015.07 passes if login fails. 8. TC_015.08 passes if login is successful. 9. TC_015.09 passes if user account is successfully deleted. 10. TC_015.10 passes if changes can be made to admin account from admin account. 11. TC_015.11 passes if admin account is successfully deleted.
TC_016	16	Select from various review categories	Uses TC_008, TC_007, TC_015 1. From no account, and from user account, select various categories and view results.	1. Passes if selected categories successfully display results.
TC_017	17	Save favorite products	Uses TC_009, TC_015 1. From a user account, save 25 products one at a time.	1. Passes if all products can be favorited, and favorites can be viewed by the user.
TC_018	18	Refresh	TC_015 1. From a user account,	1. Passes if webpage

			and from no account, refresh the webpage for home page. 2. Repeat step 1 for several product pages and category pages.	successfully refreshes and loads. 2. Passes if all webpages successfully refresh and load.
TC_019	19	Manage ads	Uses TC_015 1. From an admin account, add a new 4 new ads. 2. View the webpage with new ads. 3. Attempt to add a 5th ad to a webpage 4. Remove ads one at a time.	1. Passes if up to 4 new ads can be added to a single webpage. 2. Passes if webpage successfully displays ads. 3. Passes if error message displays when attempting to add a 5th ad to a single webpage. 4. Passes if all ads can be removed by admin
TC_020	20	Track users	Uses TC_015 1. In admin account, verify that view counter is visible. 2. Use file “IP_spoof.txt” to simulate visitors to website. 3. Verify that website views increase by 100 per iteration performed.	1. Passes if counter is visible from admin account, and not public. 2. Passes if file successfully loads and executes. 3. Passes if views increase correctly.

TC_021	21	Manage preferences	Uses TC_020 1. Load "Product_test.txt" and select a random product from the list. 2. Verify that view counter is visible on webpage. 3. Use "IP_spoof.txt" to increase view counter on product. 4. Verify that view count increased by 100 per iteration.	1. Passes if product list loads and products are viewable. 2. Passes if view counter can be seen on website, without logging in. 3. Passes if file successfully executes 4. Passes if view count increases by correct amount.
TC_022	22	Display error message dialogue box if no results found	Uses TC_002 1. On webpage, search for a non-existent product.	1. Passes if correct error message is displayed, "No products found"
TC_023	23	Display contact information	Uses TC_015 1. On webpage, verify that company contact information is accessible without an account, and from a user account.	1. Passes if contact information matches wire frame location, and is visible without being logged on, and while logged in.
TC_024	24	Load product list	Uses TC_002 1. Load file "Product_search.txt" 2. Search for the first 20 products listed at the top of the file.	1. Passes if file successfully loads. 2. Passes if number of search results

				matches number given next to product at the top of the file.
TC_025	25	Update Reviews	Uses TC_015, TC_011 1. From admin account, load file "Product_test.txt" 2. From user account, post review to any product found in file. 3. From webpage without being logged in, post review to any product found in file. 4. From admin account, delete a review added.	1. Passes if file loads. 2. Passes if database updates with new review 3. Passes if database does not add review without being logged in(TC_011). 4. Passes if review is successfully removed from both database and website.
TC_026	26	Search parameters put in place	Uses TC_002 1. From: a. Not logged in b. User account c. Admin account 1. Select search parameters given at the end of the .txt file and search	1. Passes if results match expected output given at bottom of the .txt file for all 3 use cases
TC_027	27	Store user reviews	Uses TC_015 1. From admin account, attempt to gather reviews for a product from a third-party website. 2. Verify the database	1. Passes if reviews from third-parties can successfully be gathered. 2. Passes if the

			to ensure reviews were successfully added.	database shows the correct reviews matched to correct products.
Non-Functional				
TC_028	28	View 100 reviews per page	Uses TC_002, TC_024 1. Search for the keyword specified in the .txt file 2. View the results shown on the first page. 3. Select “Show more” option at bottom of search results	1. Passes if .txt file loads successfully, and search returns results. 2. Passes if 100 products are shown on the first page. 3. Passes if the option to “Show more” is visible, selectable, and brings the user to the next results page.
TC_029	29	Display review results within .5 seconds and continue to build more	Uses TC_028, TC_002 1. Print out time taken to retrieve and display retrieved results for a product	1. Passes if time take is equal to or less than 0.5 seconds
TC_030	30	Limit searches to 50 words to provide the most accurate results	Uses TC_002 1. Attempt to search using more than 50 words	1. Passes if search bar stops accepting characters once 50 words have been

				reached.
TC_031	31	Save favorite products within .5 seconds	Uses TC_017 1. From a user account, select a product and choose to favorite it. 2. Print out time taken to return success.	1. Passes if product can be favorited. 2. Passes if time taken to return a success is equal to or less than 0.5 seconds
TC_032	32	Display category list within .5 seconds when user hovers over “Category”	Uses TC_008 1. Hover over category list with mouse	1. Passes if list is displayed within 0.5 seconds
TC_033	33	Add/delete reviews within .5 seconds	Uses TC_011, TC_025 1. Add a review. 2. Delete a review.	1. Passes if review is successfully added in 0.5 seconds. 2. Passes if review is deleted in 0.5 seconds.
TC_034	34	Log in user within 2 seconds	Uses TC_015 1. Log in to user account	1. Passes if login succeeds within 2 seconds
TC_035	35	Create accounts within 2 seconds after user provides necessary information	Uses TC_015 1. Attempt to create 1000 valid user account	1. Passes if average time for account to be created successfully is within 2 seconds of submission.

TC_036	36	Store and display user reviews within .5 seconds	Uses TC_001, TC_007 1. Run test cases 1 and 7 with timer	1. Passes if display is visible within 0.5 seconds
TC_037	37	Limit ads to a maximum of 4 per webpage	Uses TC_019 1. View every page that can be navigated to from home screen, and 20 different product pages.	1. Passes if no more than 4 ads are visible on each page visited.
TC_038	38	Display error messages within 1 second	Uses TC_009, TC_011, TC_013, TC_015, TC_022 1. Enter incorrect information and time the delay before receiving error message	1. Passes if delay is equal to or less than 1 second
TC_039	39	Display refreshed web page within 2 seconds	Uses TC_018 1. Refresh every webpage within the website and record the time taken for refreshing the webpage for 5000 times	1. Passes if average refresh time takes 2 seconds or less.
TC_040	40	Display search results using preferences within .5 seconds	Uses TC_002, 028, TC_029 1. Initiate search of a valid and existing product using the dropdown advanced search parameters and record time taken for displaying result. Perform this test 1000 times.	1. TC_040 passes if the average time takes 0.5 seconds to display search results with advanced parameters filter.

Implementation/Coding

Implementation Plan

In order to implement our solutions, we will begin developing the components that we think are more likely to encounter issues first, and save the more straight-forward development until the end.

We will start by having one team of programmers begin the program to data mine reviews, using the Splash software package, since it will be necessary to learn the software before being able to develop custom script to use with it, this will give them the most time to get the most crucial piece working. In parallel with the development, we will have teams programming the database infrastructure, using SQL, that holds the product and review information, user accounts, and the website files. Since these are vast, and will need to be tested in conjunction with the Splash program, they will be given equal priority. The data management will be implemented using MySQL, PHP and Apache; the data storage system will be implemented using IDrive. The review management system will be implemented using CMS by PHP with MySQL in order to extract product reviews from other e-commerce websites.

Once these programs are nearing completion, we will have a team of developers begin on the website interface. This is an important part of the process, but because website design has become straightforward, we will need the least amount of time building it compared to our other components. We have decided to implement our website using HTML and JavaScript to define a custom website, as opposed to a COTS solution, as we want to have full control over the layout, and full access to each component to ensure compatibility with our other components.

Potential Implementation Issues:

1. Using Splash, a COTS solution, could include extra time costs involved with learning the program, and getting it to behave the way we need.
2. Custom-building our UI instead of using a COTS solution could take longer, but we want control over every piece included.
3. Version Management will be an issue that everyone must be cognizant of. With our solution heavily relying on databases and a data-mining operation, it’s imperative that all versions be compatible and up-to-date
4. Host-Target Development will be a concern because we are expecting these databases to be held on third-party servers, and thus can only simulate the servers for our tests.
5. Open-Source or not? While Open-Source software could potentially help us with the data-mining program, at this time we feel it would be just as quick to develop our own solution using a COTS package, as it would be difficult to find one that could potentially meet our needs, as our solution has never been done before.

Testing Plan

The test cases for each component of the four main systems will be designed initially. The test data will then be prepared in accordance with each of the test cases mentioned in the test cases/RTM table. The review management website will be run with the test data that has been prepared for the test cases as inputs. The test results will be obtained as outputs and compared with the anticipated results to the test cases. Finally, the test reports will be generated.

The system will undergo **development testing** by the team responsible for system development to potentially reveal bugs and defects. This phase of testing consists of three major components:

(I) Unit Testing: Individual program units/components or object classes specified in the SE document are tested in isolation. This includes methods or functions within objects, object classes with attributes and methods, and the composite components containing defined interfaces that provides access to its functionalities. This phase will focus on testing the functionalities of objects and methods specified in the document. Object class testing is carried out in this process where all the operations associated with objects are tested along with interrogation and setup of all object attributes, and objects are exercised in all possible states. Unit testing will have two types of unit test cases: one reflects normal operation of a program and exhibit expected behavior of components, and the second one tests for proper processing of invalid and abnormal inputs where it does not crash the component being tested.

Unit testing for website components will be automated wherever applicable; for the Review Management Website, the tool Splash will be used for browser automation that retrieves and renders the web page similar to a web browser. One or multiple automation frameworks will be used to construct and run the tests. The testing frameworks should provide generic test classes that will be extended to construct specific test cases. All the implemented tests will then be run, and a report of success or failure will be generated.

The automated test components include three parts:

Setup: For this automated testing phase, the system is initialized with the test cases i.e. the expected inputs and outputs; **Call**: objects and methods are called for testing; and **Assertion:** the result of the calls are compared with the expected results. Assertion evaluation of true or false shall determine pass or fail respectively.

Inputs are chosen such that they force the system to generate all errors and causes input buffers to overflow. Identical series of/inputs are repeated numerous times and attempts will be made to generate invalid outputs while the computation results are forced to be too small or large where applicable. These processes constitute Defect testing.

(II) ***Component Testing:*** Composite components are tested in this phase and testing the component interfaces is prioritized. This phase will undergo **interface testing** to ensure that the component interfaces behave according to specifications and to detect bugs due to interface errors or assumptions about the interfaces that are invalid. The Interface types may include: *parameter interfaces*- data passed from one method/ procedure to another, *shared memory interfaces*- functions sharing the same block of memory, *procedural interfaces*- set of procedures encapsulated by sub-systems to be called by other sub-systems, and *message passing interfaces*- where services are requested between sub-systems. Interface misuse, interface misunderstanding, and timing errors are possible factors that will be observed.

Plan: Tests are designed where the parameters for the called functions are set at the extreme end of their ranges. Component failure is tested with tests designed with intent to fail. It is followed by **stress testing** with the system associated with message passing, and for systems where memory is shared, the order is varied in which the components are activated.

(III) ***System Testing:*** Components will be partially or completely integrated including the off-the-shelf system associated with data mining, and the system will collectively undergo testing at its entirety. It will focus on the interaction between components, discover bugs in the system in the form of **defect testing**. It will also verify compatibility of components, timely transfer of appropriate between interfaces, and test the emergent behavior of the systems. Given the size of the team, system testing will not be done by a separate testing team.

The test cases for Account, userAccount, adminAccount,… are defined. Using the state model in SE Document, sequences of state transitions to be tested and event sequences that triggers these transitions are identified.

Use-case testing*:* System testing will be based on the use-cases provided in the engineering document as it identifies the system interactions. The test cases are defined in the document and these cases may involve multiple system components and testing the use cases hence forces interaction between these systems, for example, creating user account with user information and successful login using valid login parameters ensures successful retrieval of information from the databases in the server/data storage system. The sequence diagram associated with the use cases provides documentation of the components and interactions that will be tested. All input requests render associated acknowledgements. Summarized data is created to check for correct organization of the report, eg: input request for report to “Account” should generate a summarized report; raw data can be used to test objects and to test the generated report. Testing policies defining the required test coverage are complied with.

Since this project is not adhering to Agile methods, test-driven development approach is not considered for this website development plan as the cons outweigh the pros and regression testing associated with test-driven approach might increase costs. Requirements based testing is followed for this project where tests have been developed upon examination of each requirement and usage scenarios are considered for the website.

The website will undergo ***release testing*** by a separate testing team, outside of the development team, that will test a complete version of the website before being released to the users. The goal is to show that the system delivers its intended functionalities, assess reliability and performance, and to get system supplier’s approval for use upon passing the validation testing phase. In this black-box system testing process, the tests will only be derived from the system specifications and it should reflect the profile use of the system. To test the system for meeting performance (non-functional) specifications, series of tests are conducted where the load is steadily increased until the system performance is unacceptable. Stress testing is done to test for performance conformity by deliberately overloading the system to the point of failure.

User testing will be undertaken as a final step where the potential users of the website will test the system in their own environment. These customers will then provide input or feedback and advice on system testing. The types of user testing include: Alpha testing, Beta Testing, and Acceptance Testing. During **Alpha testing**, the users will work with the development team at the developer's environment to test the website. It will be followed by **Beta testing** where a release version of the website will be made available to the users for them to use the full-fledged website and provide feedback or bug reports to the development team.

Finally, it advances to ***Acceptance Testing*** where the customers will test the system and decide to accept the website project from the system developers or not. If accepted, the Review Management Website will be deployed in the customer environment primarily for custom systems.

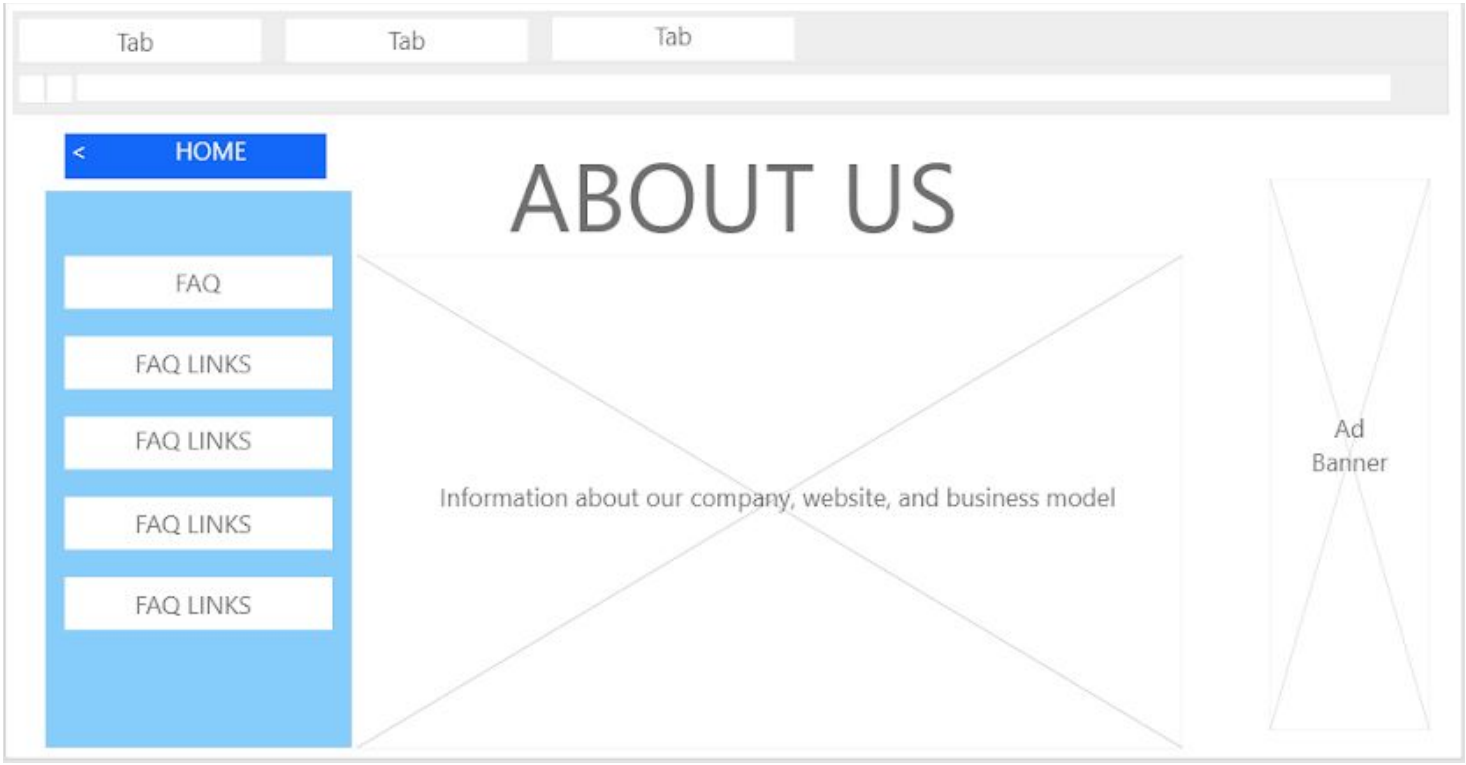
For the acceptance testing phase, the plan is as follows: acceptance criteria is first defined, acceptance testing is planned, acceptance tests are derived and run, the test results are negotiated, and a decision is made on accepting or rejecting the system.

Potential Testing Issues:

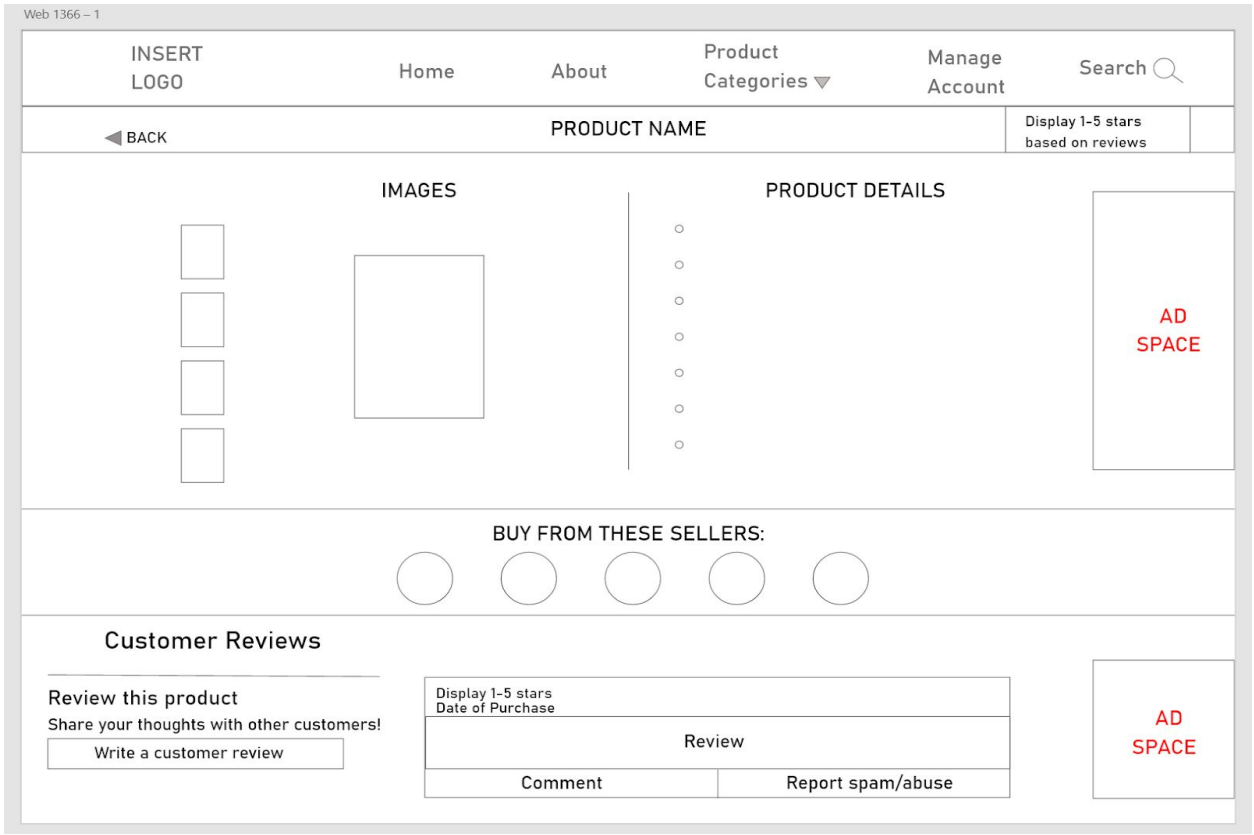
1. It can be difficult to simulate all possible web-client platforms, environments, device specifications, and scenarios while testing, and some of these web clients may not be able to handle some of the content returned by the server.
2. Session problems or concurrency issue might arise when multiple users are active on the same page and/or when an individual user is active on multiple windows of the same page.
3. Advertisements and COTS applications may not function or integrate as expected and may cause irregular behavior with associated systems that might make it difficult for testing.
4. During testing, some errors can mask other errors and go undetected.
5. Inheritance can make it difficult to design object class tests as the information to be tested is not localised.
6. Broken links.
7. Cookies may fail to work in some systems.

Wireframes

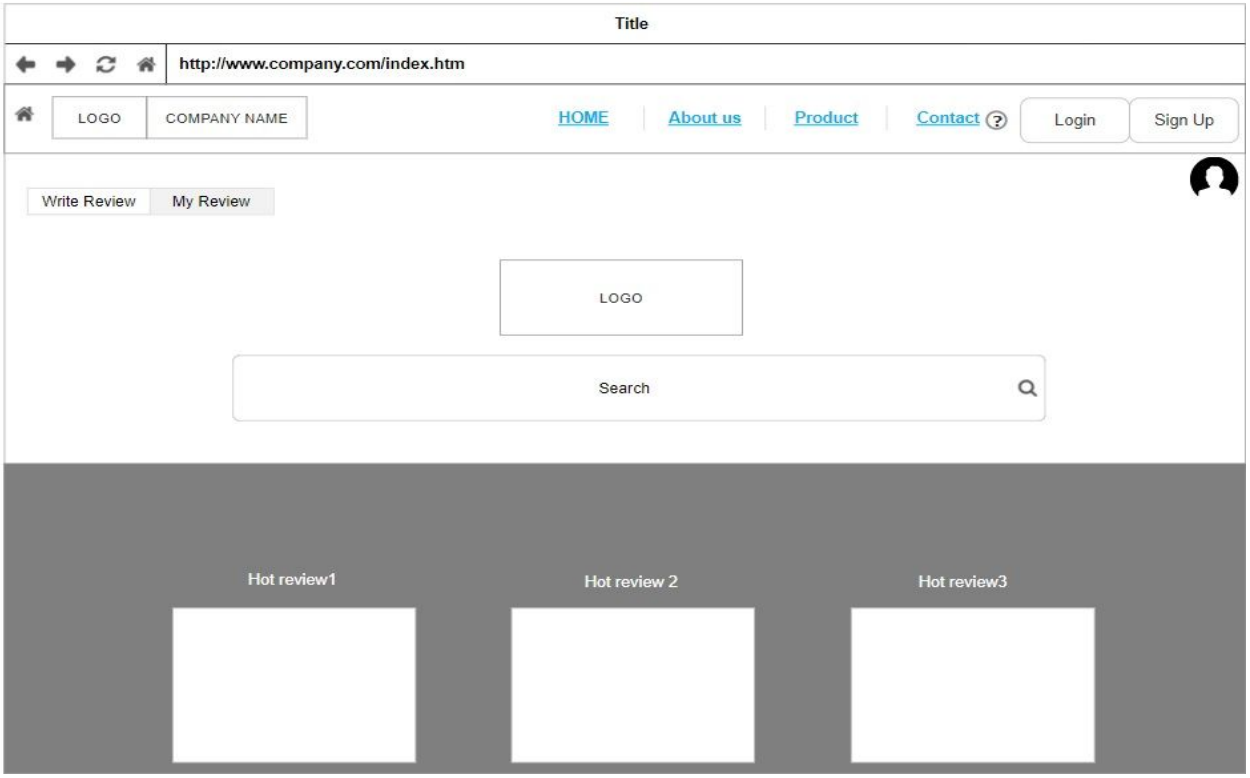
About Page



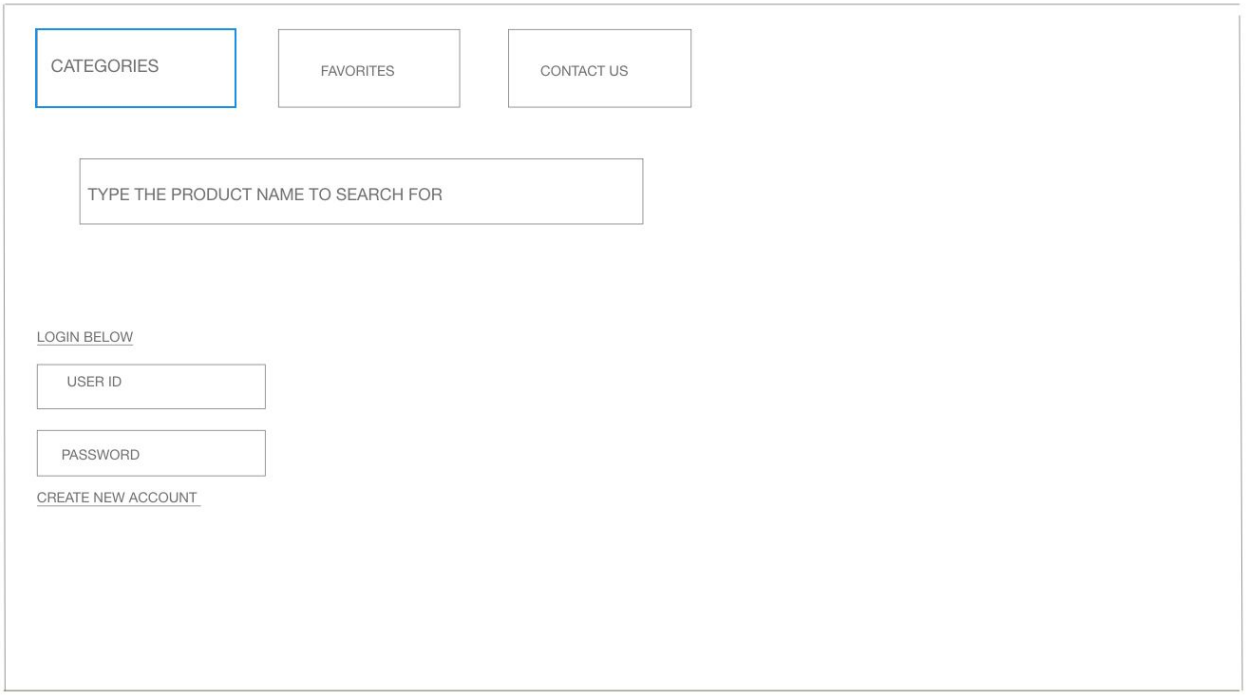
Product Review Page



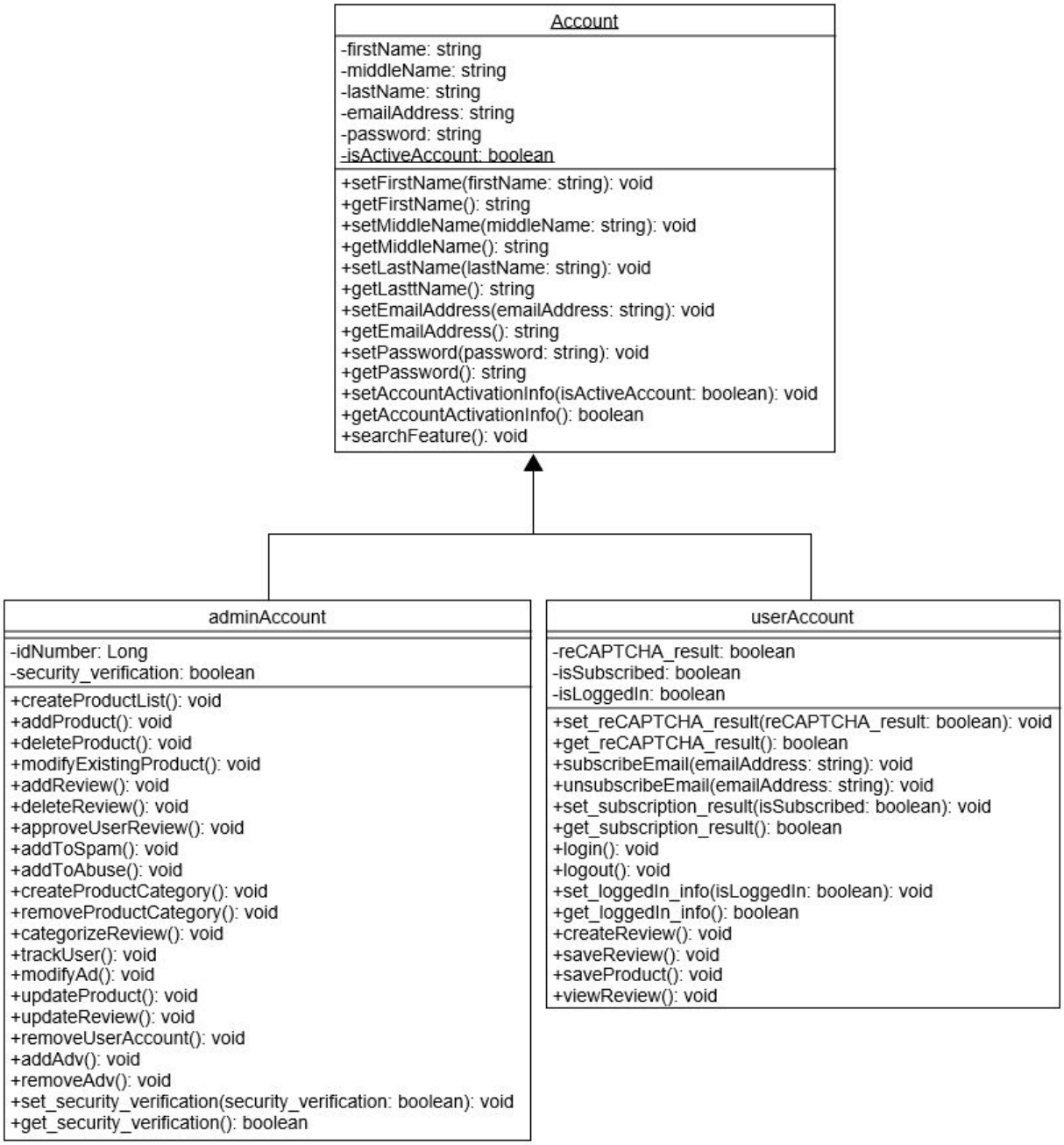
Home Page


Title
http://www.company.com/index.htm
LOGO
COMPANY NAME
HOME
About us
Product
Contact
Login
Sign Up
Write Review
My Review
LOGO
Search
Hot review1
Hot review 2
Hot review3

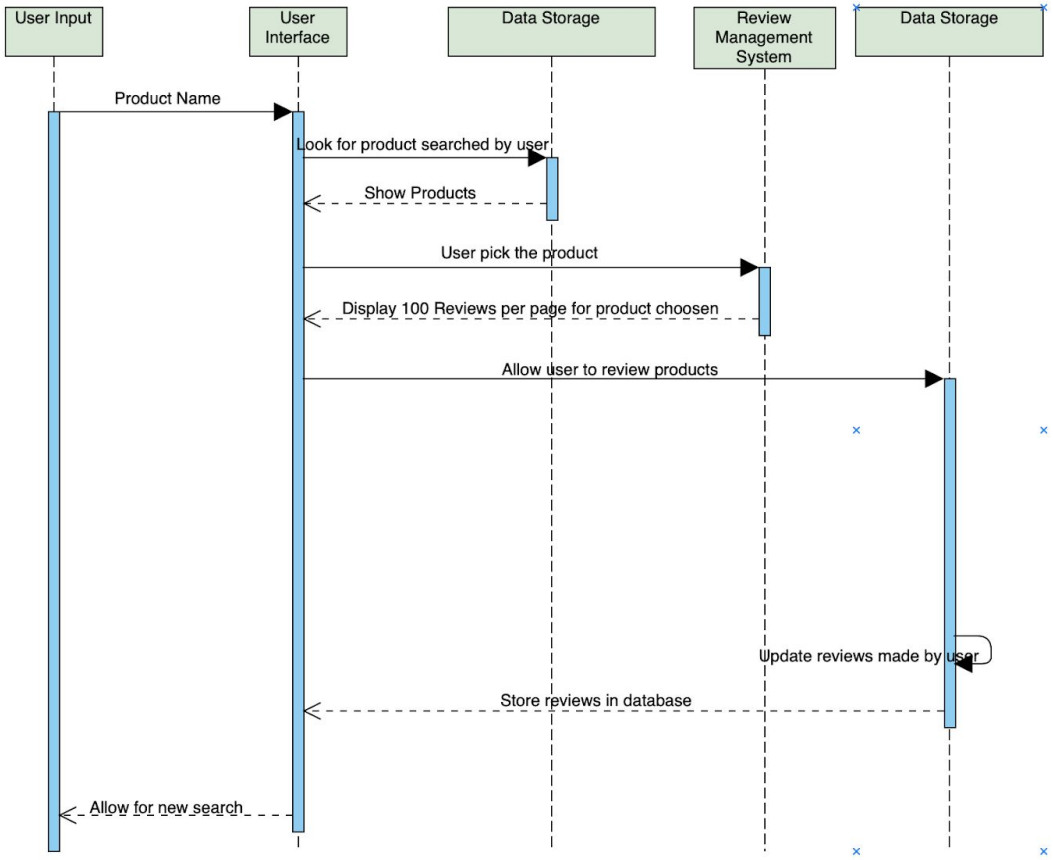
Login Page


CATEGORIES
FAVORITES
CONTACT US
TYPE THE PRODUCT NAME TO SEARCH FOR
LOGIN BELOW
USER ID
PASSWORD
CREATE NEW ACCOUNT

Object Classes



Sequence Diagram



Deployment

The website will be deployed through cloud servers and databases purchased by the client, as defined in the Statement of Work. The client will be responsible for registering the domain name and purchasing/providing web hosting service. Before the website is set up on the host service, there must be access to DNS management administration. The website must be deployed first on a live server environment. The deployment team can set folder permissions and other settings, and then run tests and benchmarks to see how the site performs on the server. Before website deployment on a live server environment, take a full backup including databases as well as directories.

Maintenance

First step in maintenance is the creation of databases and website file backups; do not back up on the same server in case of server failure. Install latest updates for the website software, themes, and plugins or modules. Check the website for broken links using link checker tools online. All forms and fields must be tested on the website; this includes, but is not limited to, contact and email list sign up forms, and site search fields. Furthermore, the functionality of the product review system must be constantly tested and maintained. The search engine must also be tested and maintained; large jumps and overall trend must be accounted for, with the help of an Internet marketing consultant. Addition of new content must be scheduled for updates; the schedule should serve as the maximum time period that may acceptably pass between updates.

Lessons Learned

1. Setting a recurring meeting time, and having a majority of the group attend is crucial to the success of a project.
2. Delegate tasks to group members who are *capable* of handling them.
3. As important as project management is, it is crucial that other team members listen to what the project manager recommends; team members must also be open to receiving constructive criticism when necessary.
4. The initial idea of the solution can shift and evolve during the life cycle, and it's important to go back and update the documents that describe the solution.
5. Without clear requirements, everything that follows in the development life cycle becomes more difficult.
6. It is important to assess risks and allocate more resources in areas of the system that are not familiar and may have more risk.
7. Project deliverables should be completed ahead of deadline and the work should be verified with the team to ensure that it aligns with project requirements before being consolidated in submissions.
8. Defining system constraints and out-of-scope is a crucial step in the development life cycle of a system that is large and complex.
9. Any assumptions made should be clearly defined in the Statement of Work as well as SE document.
10. For a programmer, reliable documentation is always necessary. The presence of documentation helps keep track of all aspects of an application and it improves on the quality of a software product.