

CS 5200 PROJECT : IDP ELECTRONIC SHOPPING WEBSITE REPORT

TEAM: DCODERS

**TEAM MEMBERS: ISHANI DOSHI, DHWANI SONI, PRANEETH
PULLIGUNDLA**

Problem statement

The goal was to create a shopping website where a user can buy electronic items like Laptops, Tablets, Mobile phones and Desktops.

Details:

We planned to fetch the items from an API and show it to the users of our website.

The items purchased by any registered user through our website will be stored in our database.

Comments added by users, products saved by the user will also be saved in our database.

Users profile which includes his/her address, credit card details along with the comments he/she made, products he/she saved, and his/her purchase history all will remain in our database which he/she can update/delete as required and as authorized.

There will be an admin of the website who only will have privileges to generate report based on users and products in our database.

Proposed solution

- Using eBay's Finding API and Shopping API, we are fetching items from eBay given a **keyword or list of keywords** to finding API and **item Id** to shopping API.
- There can be 3 types of users of this website:
 1. Guest User
 2. Registered Member
 3. Admin

Guest user can only search the products and can see particular product's details.

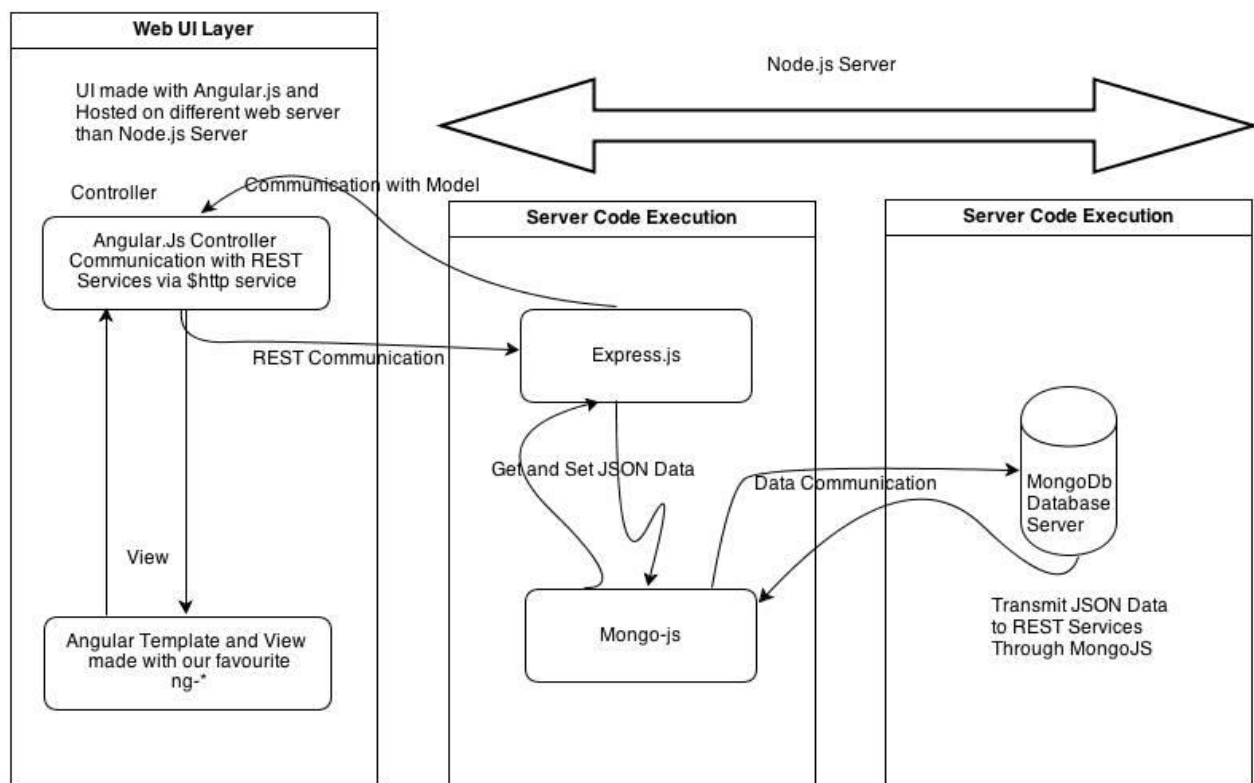
Registered members can do whatever guests can do and additionally they can make a purchase, save product for later, give comments to a product, view his/her purchase history, view his/her profile and other member's profile.

Admin can generate reports based on a given user, all users, given product, all products that are saved or purchased by a user.

- We are creating a MEAN stack application to implement this.

Architecture & Technology stack

- APIs used :
 1. eBay's Finding API
 2. eBay's Shopping API
- Technology stack used: (MEAN stack)
 1. MongoDB
 2. Express web service
 3. Angular JS
 4. NodeJS



APIs

We have used 2 APIs in this project

1. **EBay's Finding API** : Given a keyword or list of keywords, we can fetch list of products with its details from eBay.
2. **EBay's Shopping API** : Given an item id, a particular product can be fetched from eBay.

Use case analysis

1. **Use Case:** Create new user [SignUp]
Description: User wants to create his profile and use the website for uses other than just searching products.
Actors: User
Steps:

Actor Actions	System Responses
Enters Username/Password, security question, answer, username, password	Validates entered information
Submit	Creates User Profile

Alternate path

Username already exists	Error: Username already exists. Select other username.
Mandatory fields are not filled	Error: Please fill all mandatory fields.
2. **Use Case:** Forgot Password [forgotPassword]
Description: User had forgotten his password
Actors: User
Preconditions: Username should prevail in System.
Steps:

Actor Actions	System Responses
Answers security Question	Provides reset password page if answers match to the saved data.

Alternate path

If security answers don't match.	Would not allow to reset password
----------------------------------	-----------------------------------
3. **Use Case:** View Comments
Description: User wants to view his/her existing comments.
Actors: User
Preconditions: User should be authenticated
Steps:

Actor Actions	System Responses
Click View Comments on my profile page.	Shows all the comments a user has made till now.

Redirects back to profile

Description: User wants to add comment to any product.

Preconditions: User should be authenticated

Actor Actions

System Responses

description page.

and shows new added information on the page.

Description: User wants to edit comments in his profile.

Preconditions: User should be authenticated

Actor Actions

System Responses

edit it

Redirects back to profile

Description: User wants to delete his/her existing comment.

Preconditions: User should be authenticated

Actor Actions

System Responses

Redirects back to profile

Description: User wants to view his/her existing Payment Info.

Preconditions: User should be authenticated

Actor Actions

System Responses

on my profile page.

Redirects back to profile

Description: User wants to add new payment info in his profile.

Preconditions: User should be authenticated

Actor Actions

System Responses

Add new credit card

Done	and shows new added information. Redirects back to profile
------	---

9. **Use Case:** Delete Payment Info

Description: User wants to delete payment info in his profile.

Actors: User

Preconditions: User should be authenticated

Steps:

Actor Actions	System Responses
---------------	------------------

Delete already added credit card	Update information about user in database and shows new updated information.
----------------------------------	--

Done	Redirects back to profile
------	---------------------------

10. **Use Case:** View Addresses

Description: User wants to view his/her existing Addresses.

Actors: User

Preconditions: User should be authenticated

Steps:

Actor Actions	System Responses
---------------	------------------

Click Add/Edit Address on my profile page.	Shows all the Addresses a user has added till now.
--	--

Done	Redirects back to profile
------	---------------------------

11. **Use Case:** Add Addresses

Description: User wants to add Addresses info in his profile.

Actors: User

Preconditions: User should be authenticated

Steps:

Actor Actions	System Responses
---------------	------------------

Add new Address	Adds information about user in database and shows new added information.
-----------------	--

Done	Redirects back to profile
------	---------------------------

12. **Use Case:** Delete Addresses

Description: User wants to delete payment info in his profile.

Actors: User

Preconditions: User should be authenticated

Steps:

Actor Actions	System Responses
---------------	------------------

Delete already added credit card	Update information about user in database and shows new updated information.
----------------------------------	--

Done	Redirects back to profile
------	---------------------------

13. **Use Case:** View Saved Products

Description: User wants to view his/her existing saved products.

Actors: User

Preconditions: User should be authenticated

Steps:

Actor Actions	System Responses
---------------	------------------

Click View Saved products on my profile page.	Shows all the products saved by user till now.
---	--

Done	Redirects back to profile
------	---------------------------

14. **Use Case:** Delete Saved Products

Description: User wants to view his/her existing saved products.

Actors: User

Preconditions: User should be authenticated

Steps:

Actor Actions

Click View Saved products
on my profile page.
Done

System Responses

Shows all the products saved by user till now.
Redirects back to profile

15. **Use Case:** View Purchase History

Description: User wants to view his/her existing purchase history.

Actors: User

Preconditions: User should be authenticated

Steps:

Actor Actions

Click View purchase history
on my profile page.
Done

System Responses

Shows all the purchase history of user till now.
Redirects back to profile

16. **Use Case:** Search Product [searchProduct]

Description: User wants to search for products

Actors: User

Steps:

Actor Actions

Enters keyword to search

Alternate path

No products are found

System Responses

Displays all search results.

Displays error message showing zero results found.

17. **Use Case:** View Product Details [viewProduct]

Description: User wants to view product details

Actors: User

Preconditions:

Steps:

Actor Actions

Clicks on product
from search results

Alternate path

Clicks on product
from saved products

Clicks on product
from shopping cart or
purchase history

Clicks on product
from my comments page

System Responses

Displays that particular product details and related comments.

Displays that particular product details and related comments.

Displays that particular product details and related comments.

Displays that particular product details and related comments.

18. **Use Case:** Add to shopping cart

Description: Authorized user can add a product to the shopping cart which he would like to purchase

Actors: User

Preconditions: User authenticated

Steps:

Actor Actions

User adds a product in shopping cart by clicking on add to cart button on product description page.

System Responses

Product gets added to the shopping cart of the user.

19. **Use Case:** Remove from shopping cart

Description: Authorized user can remove existing product from his/her shopping.

Actors: User

Preconditions: User authenticated

Steps:

Actor Actions

User removes a product from shopping cart by clicking on remove button in the cart.

System Responses

Shopping cart gets updated with the product removed.

20. **Use Case:** Add to saved products

Description: Authorized user can add a product to the saved products which he would like to purchase later.

Actors: User

Preconditions: User authenticated

Steps:

Actor Actions

User adds a product in saved products by clicking on save this button on product description page.

System Responses

Product gets added to the saved products of the user.

21. **Use Case:** Payout

Description: User wants to pay and make order.

Actors: User

Preconditions:

1. User Authenticated

2. There should be items in shopping cart

Steps:

Actor Actions

Clicks payout

System Responses

Removes items from cart. Adds the products in purchase history of that user. Redirects to Homepage.

22. **Use Case:** View another member's profile

Description: User wants to view another member's profile

Actors: User

Preconditions:

1. User Authenticated

Steps:

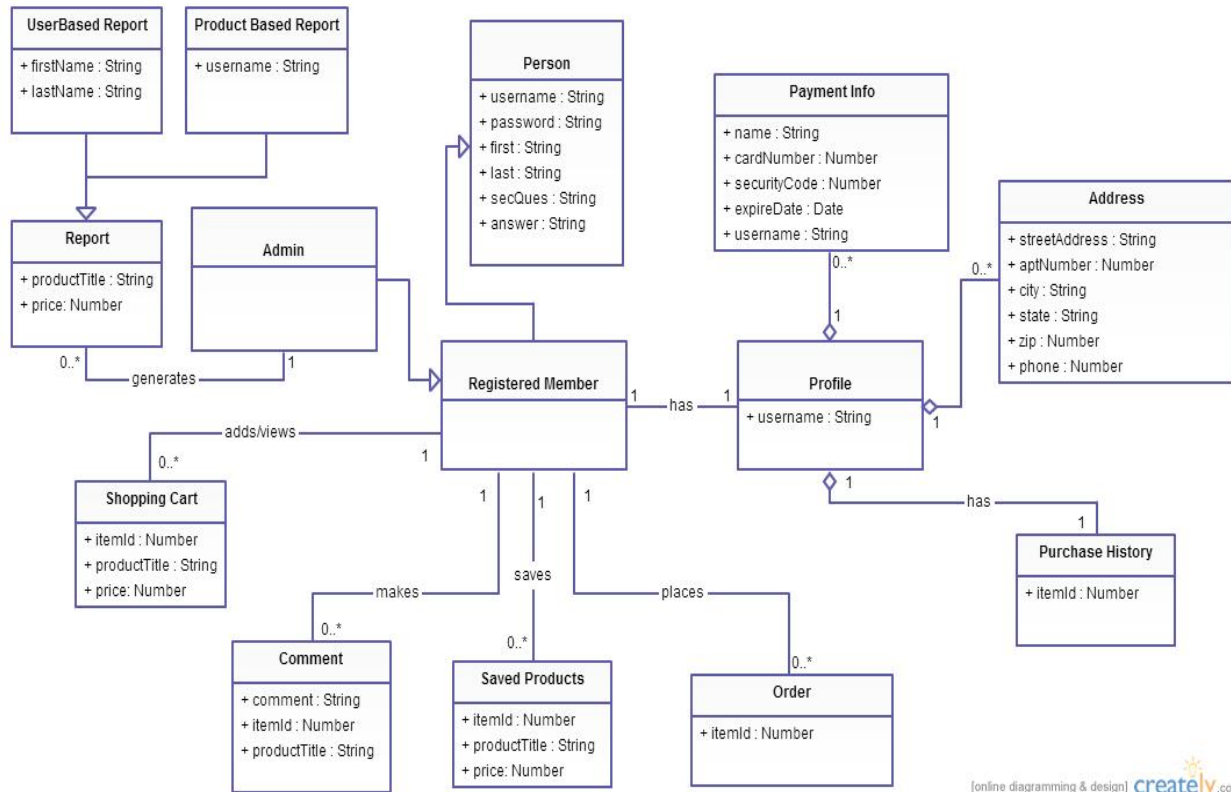
Actor Actions

Clicks on other person's username on reviews tab in product description page.

System Responses

Redirects to the profile page of clicked user showing his/her comments and saved products.

Data model



[online diagramming & design] createby.com

In a MEAN stack application, there is no need of class diagram.

But we made a class diagram to refer as a base to implement the MEAN stack application.

Information to run this website on local machine

- Please start the node server using following command on the terminal:

node ./server.js

- Visit following url to use the website:

<http://localhost:3000/>

It will land you to the sign in page. You can than start using our website.

- Admin can login using following credentials:

username: **admin**

password: **admin**

But first he needs to register if he is not already registered.