

SW Engineering CSC648/848 Spring 2020

Milestone 02

CsGetDegrees.com

Section 02

Team 04

Team Members:

1. Team lead: Anh Le - email: ale4@mail.sfsu.edu
2. Front end lead: Mohamad Farah
3. Front end developer and Github Master: YeeJian(Jeans) Tan
4. Back end lead: Aramis Knox
5. Back end developer: Timothy Wells
6. Back end developer: Russell Azucenas

History Table

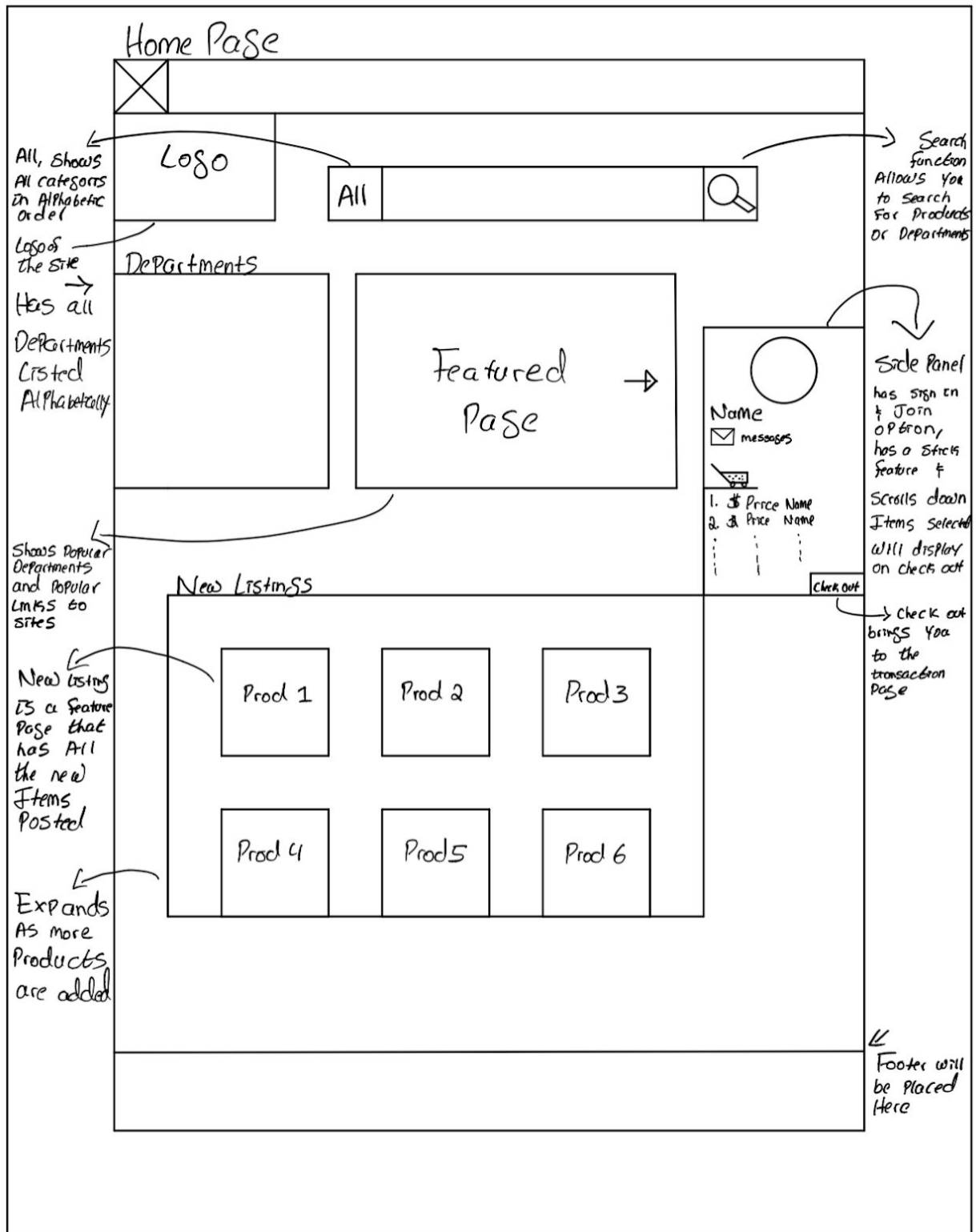
Date	Details
Mar 13, 2020	Submitted Milestone 2 to professor
TBA	

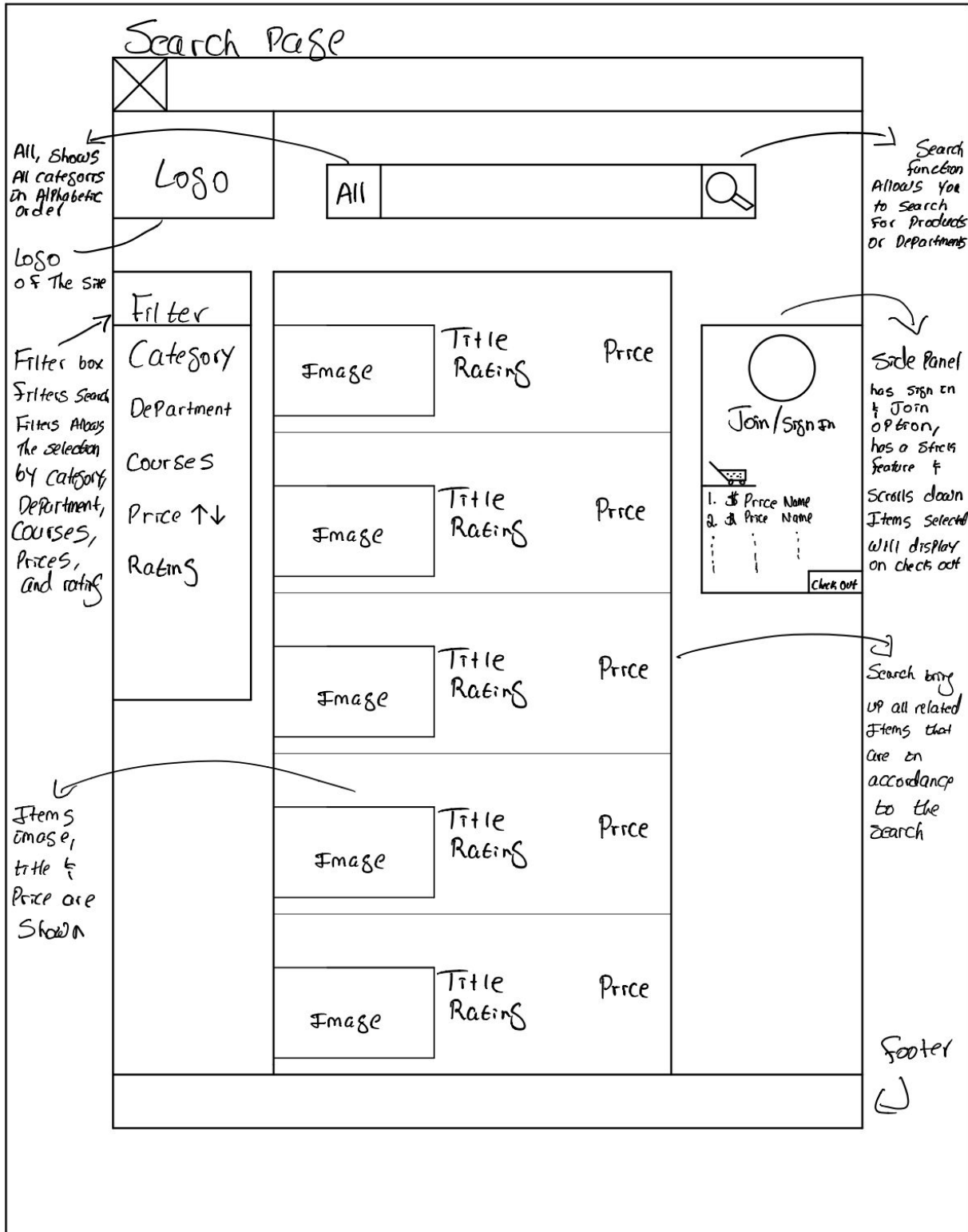
I. Functional Requirements

No.	Functional Requirement	Priority
1.	System <i>shall</i> allow approved users to buy products	1
2.	System <i>shall</i> allow approved users to buy services	1
3.	System <i>shall</i> allow approved users to sell products	1
4.	System <i>shall</i> allow approved users to sell services	1
5.	System <i>shall</i> allow all general users, to become approved users via a login	1
6.	System <i>shall</i> allow all users, even general users, to browse through the site	1
7.	System <i>shall</i> allow all users, even general users, to search through the site using a search bar	1
8.	System <i>shall</i> allow approved users to send messages to sellers through their listings	1
9.	Users that have received a message <i>shall</i> be able to reply	1
10.	Users that have become part of an initiated conversation <i>shall</i> be able to view the conversation	1
11.	Admin <i>shall</i> be able to add departments to the database	1
12.	Admin <i>shall</i> be able to add courses to the database	1
13.	Admin <i>shall</i> be able to approve product listing	1
14.	Admin <i>shall</i> be able to remove product listing	1
15.	Admin <i>shall</i> be able to remove approved users	1

16.	Products <i>should</i> have tags that can be used to enhance the search experience on the site	2
17.	System <i>should</i> allow users to review sellers after a transaction has been completed	2
18.	All users <i>should</i> be able to view other users' profiles via their product listings	2
19.	All approved users <i>should</i> have a profile	2
20.	Approved users <i>should</i> have a balance of fake currency that can be used to demonstrate how transactions would take place if a payment api were used.	3
21.	System <i>should</i> put well-reviewed users on the "Dean's List" to indicate that they are trustworthy sellers	3

II. UI Mockups and Storyboards (high level only)





Transaction Page



Final Product on the transaction page

Product 1
Description

Price \$

Product 2
Description

Price \$

Confirmation on Purchase

Confirmation

Your Balance:
Cost:

Confirmation

Email:

Confirm Purchase

1. \$ Price Name

2. \$ Price Name

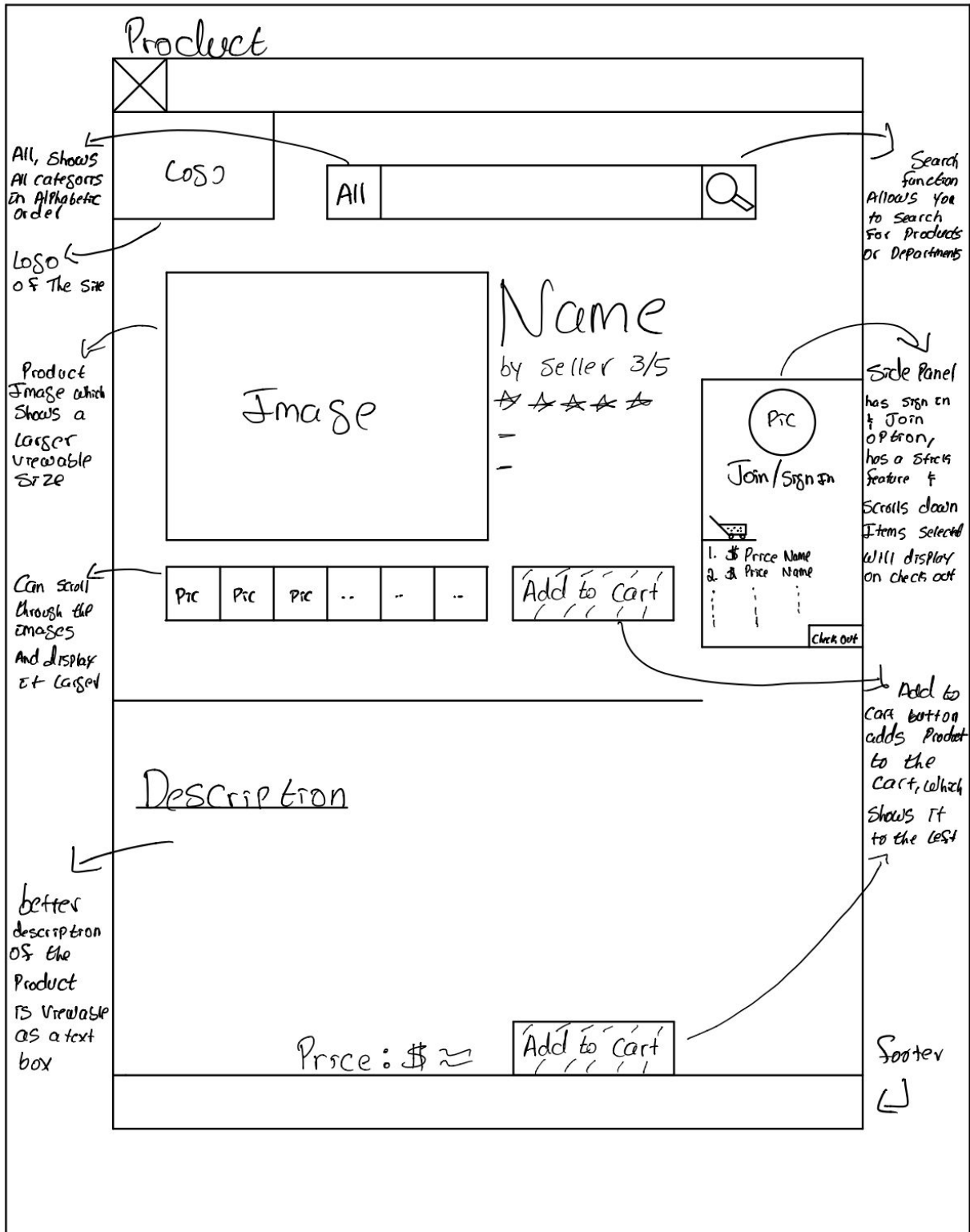
...

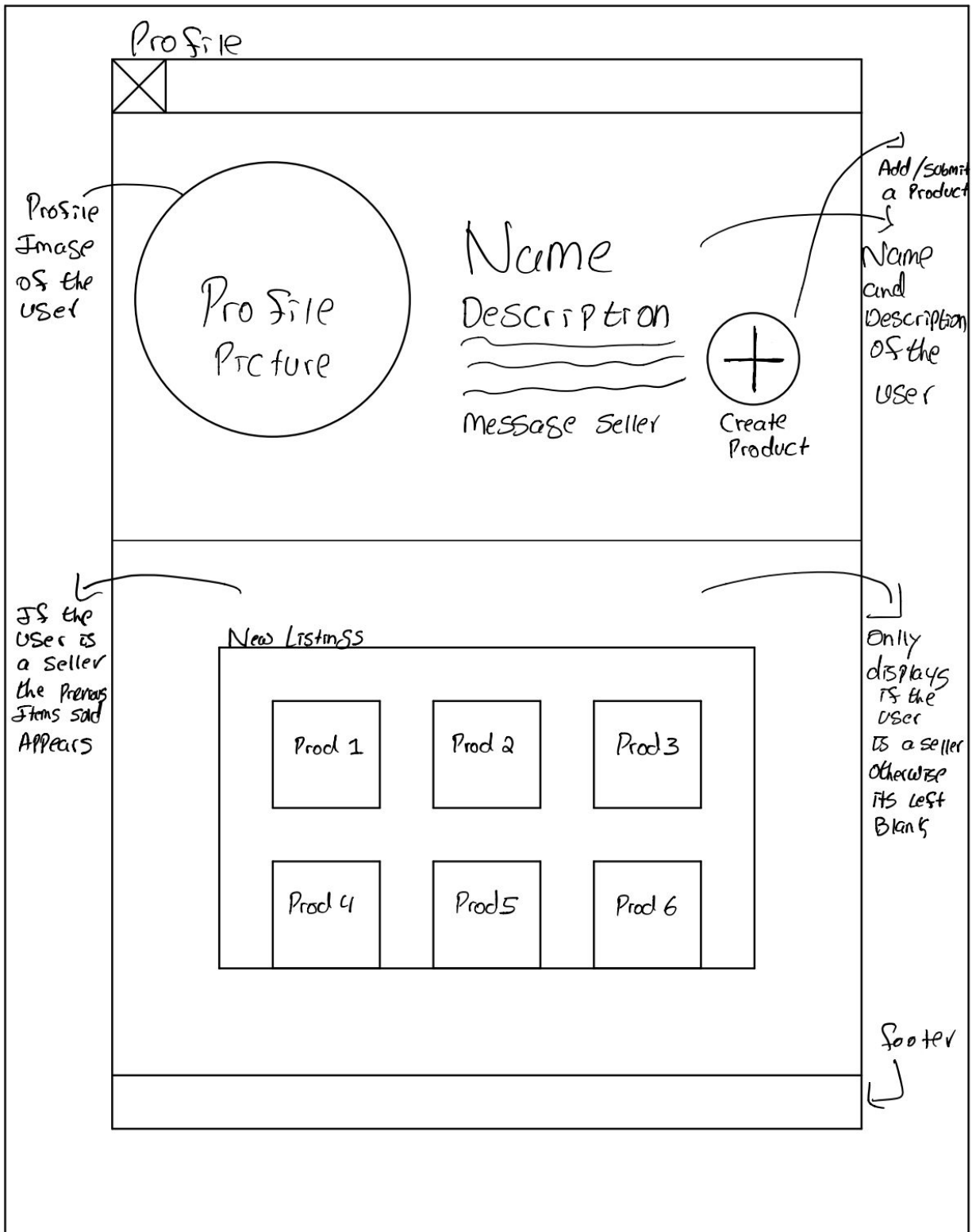
...

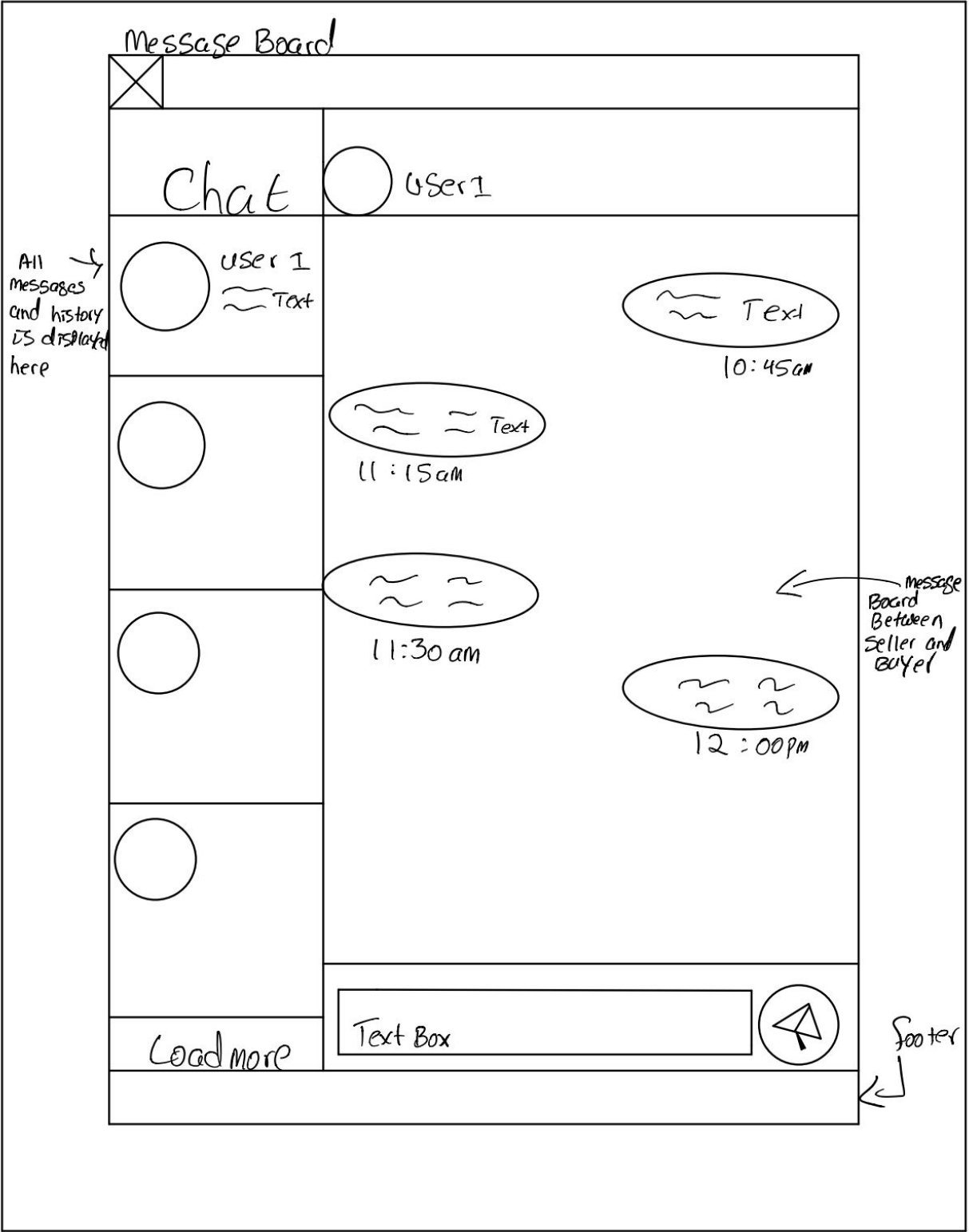
check out

Side Panel has sign on & Join option, has a sticky feature & scrolls down Items selected will display on check out

Footer







POP UP For Login

Log In

Log in
for
users
with
an
account

Username:

Password:

Create
Account

Create
an account
to purchase
items

Email:

Username:

Password:

Confirm Password:

Enter
Address

Entering
Address
for delivery

Street Address

Street Address

County / State

Postal

Country

POP UP / Create Product

Description
of the
Product

Title:

Quantity:

☐

check is unlimited

Categories:

Department:

Courses:

Tags:

Description:

Price:

☐ Agree to Terms & Agreement

Confirm

Admin Page

Product Table

→
Product
Search

id	Name	Dep	course	Cat.	Price	Approve

User List

Enter

←
search
for
user

→
user
search
result

User:

Username:

Email:

Address:

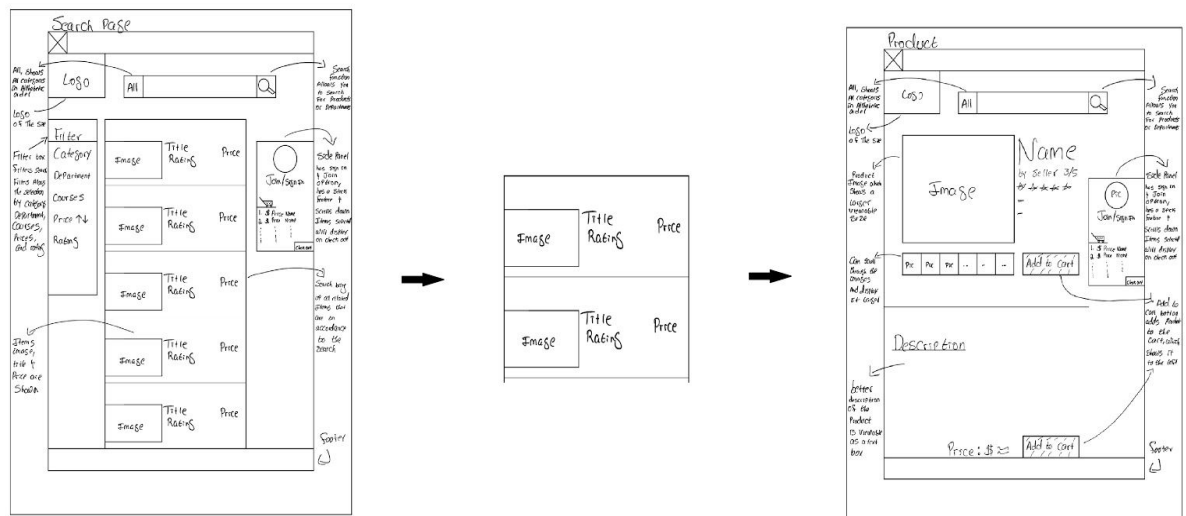
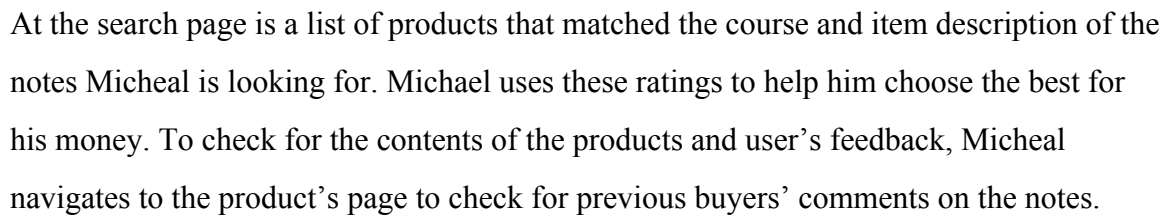
→
Product
List
of user

id	Name	Dep	course	Cat.	Price	Approve

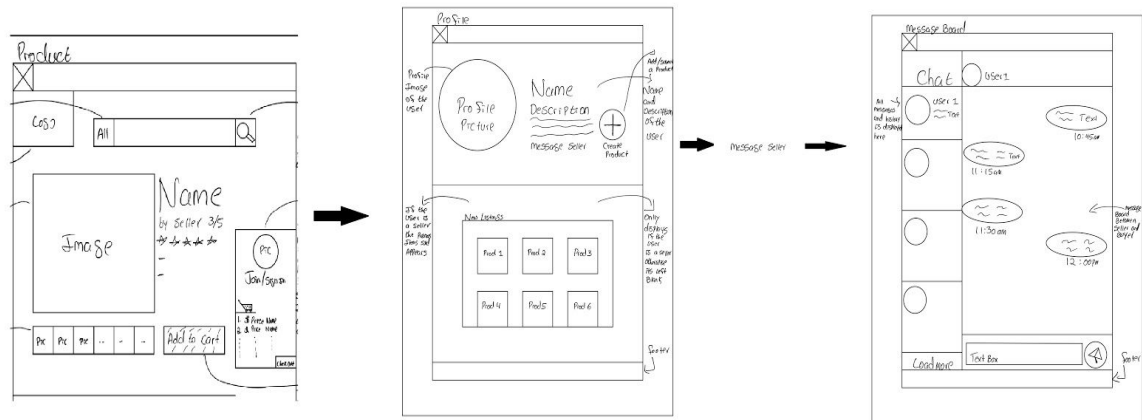
Delete User

←
copy
listed
Product
from
user

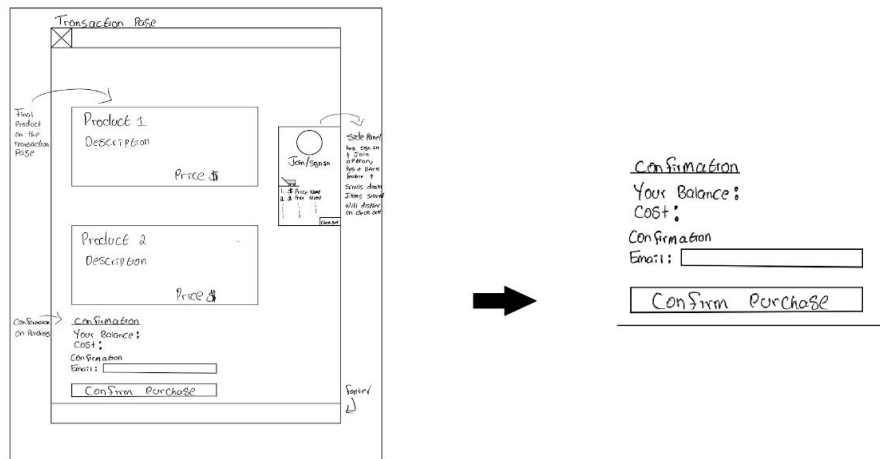
Micheal navigates to the homepage of CsGetDegree. Using the search bar, noticeably at the top of the page, Micheal proceeds to look for items he's looking for. In this case he looked up notes for his course.



To negotiate for the price of the item Michael wants, he navigates to the profile page of the seller and clicks on “message seller” to contact and ask if he seller can lower the price to a reasonable amount for them.

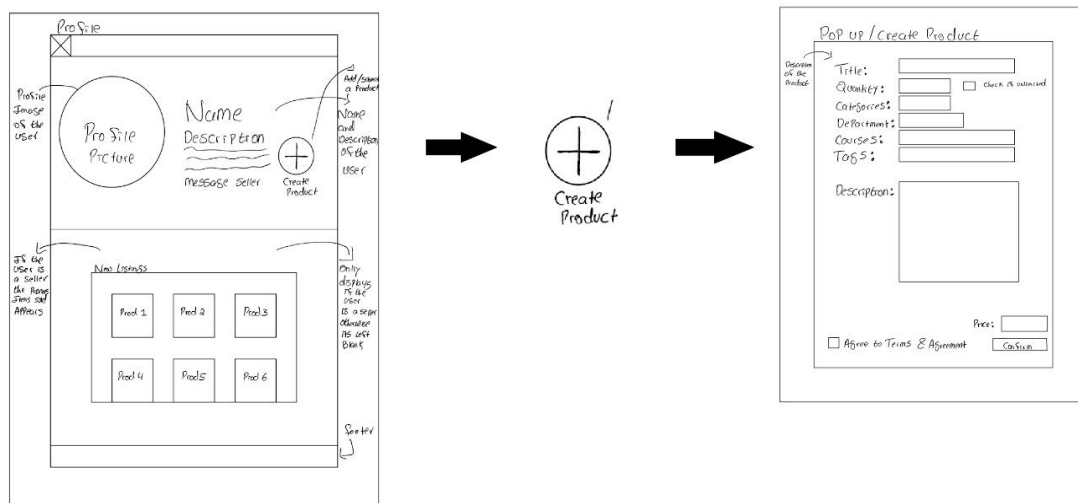


At the end of using the CsGetDegrees website, Michael walks away with a quality set of notes and a quality worthy tutor. He clicks on confirm purchase and purchases his items in the cart.



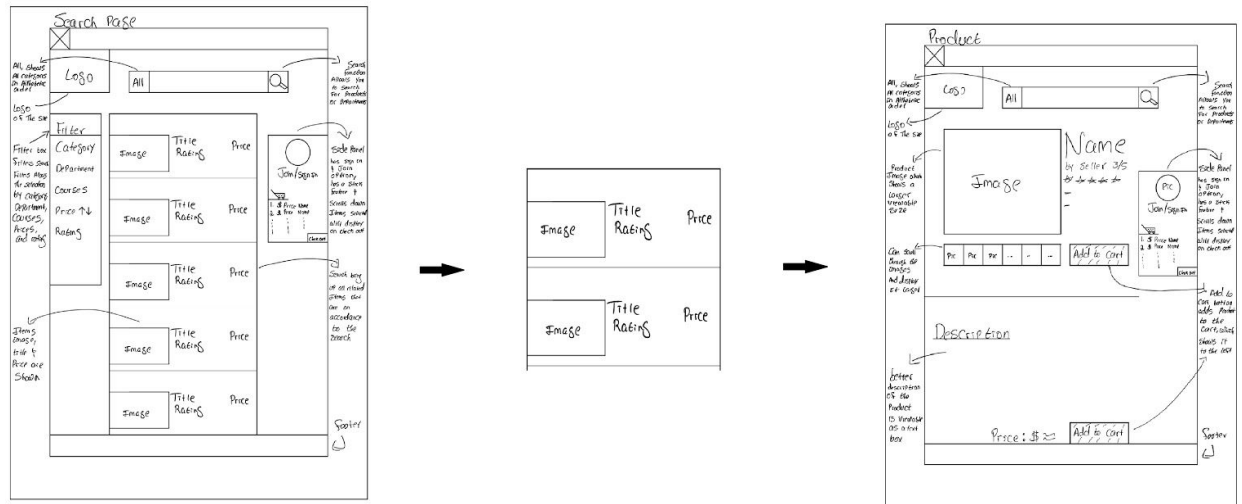
Gabriella

Gabriella wants to sell her tutoring service as a product on CsGetDegree. She navigates to the profile page and clicks on “Create product.” A pop up will appear allowing Gabriella to fill in information about her product.



In the description of the product she can list out her available hours and courses she can cover.

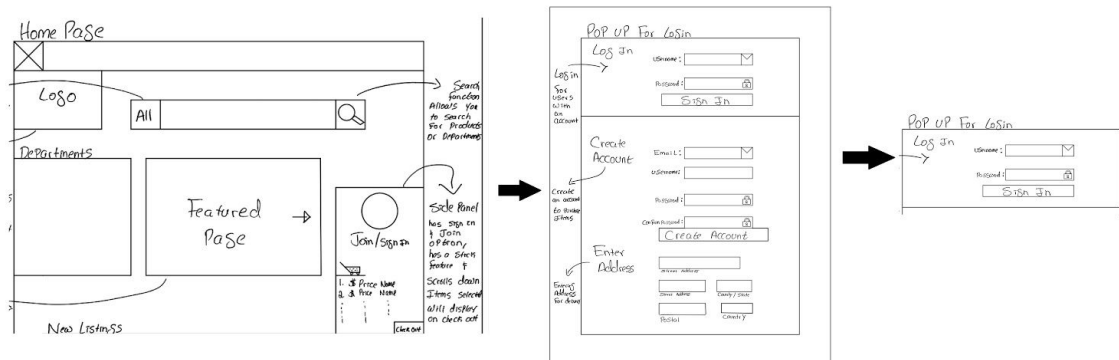
Users who want to buy tutoring as products can find Gabriella’s items on sale. With positive rating, users will be more at ease purchasing tutoring service from Gabriella.



This provides Gabriella a quick source of income by selling good tutoring services as items for users to easily find and trust.

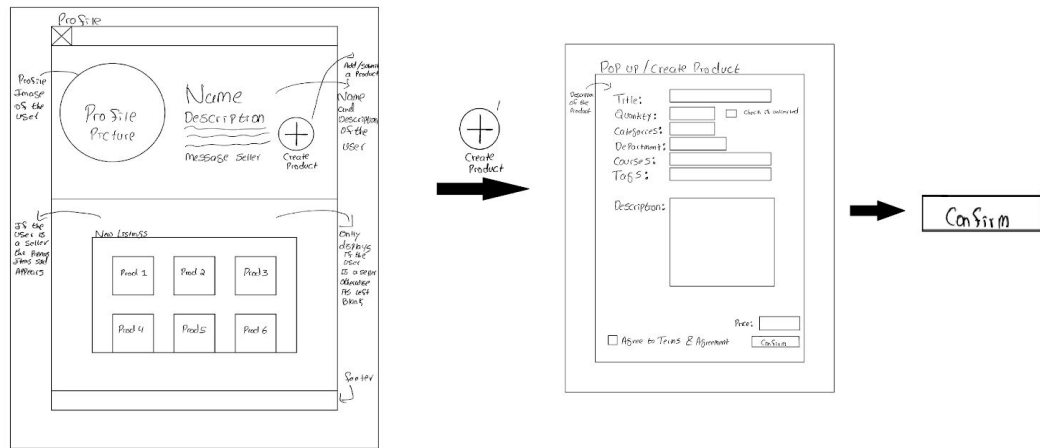
Jacob

To sell a product you must be a user. Navigating to CsGetDegree Jacob will be prompted to make an account or log in. Jacob finds it's easy to locate and navigate to the login option that pops up right away on the right side of the screen.



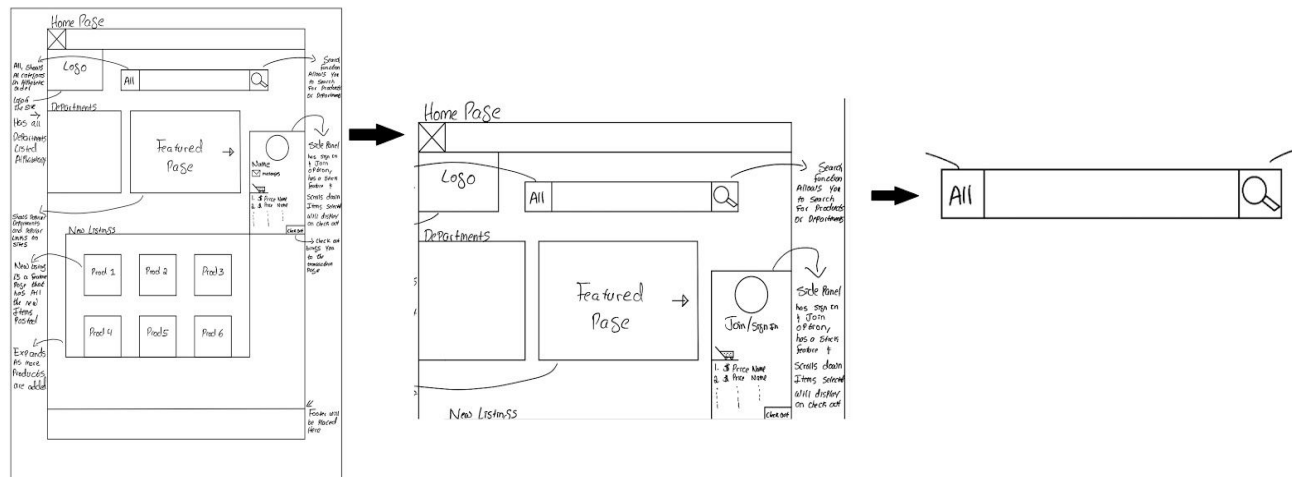
On his profile page, he can find a big plus symbol labeled “create product”, which prompts a pop up that requires Jacob to fill in a few information about the item he’s

selling. After filling out the form, Jacob is one click away from easily selling his item he's been wanting to sell but didn't know how to.



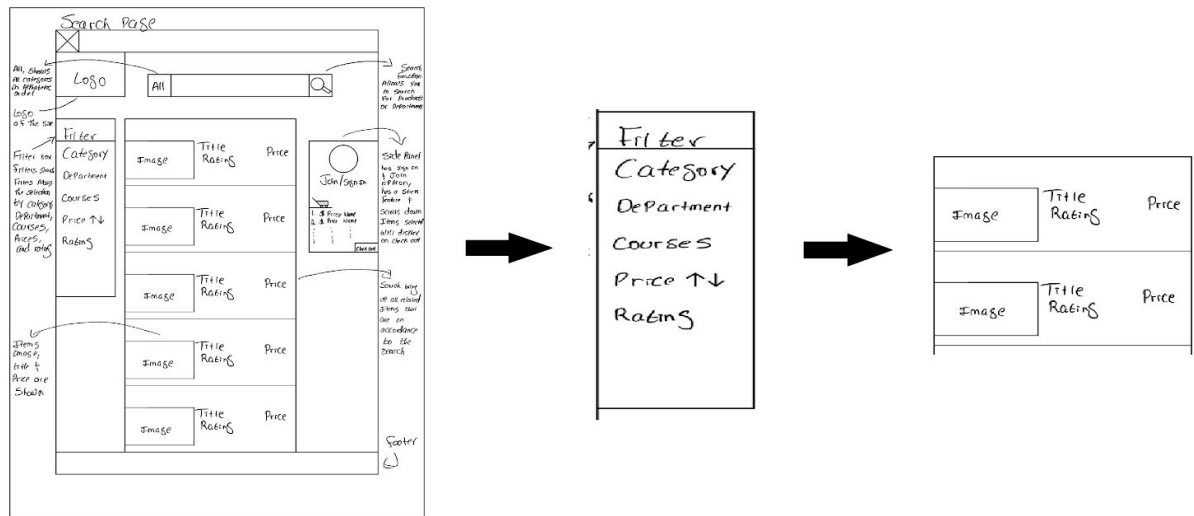
Christina

Christina can search for babysitter services as purchasable items in the search bar located in the home page.

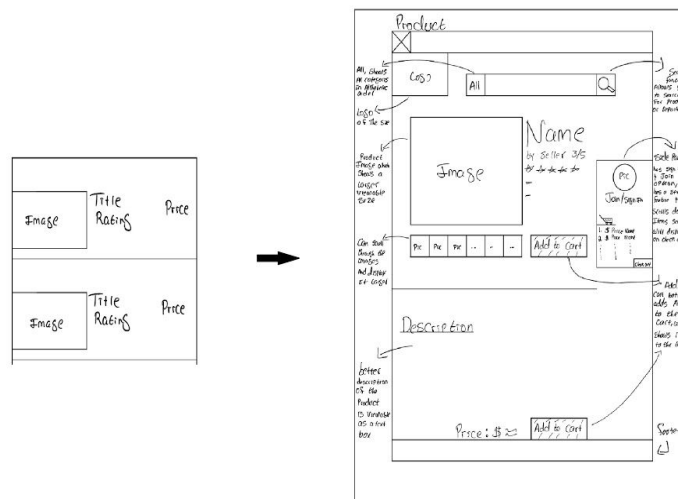


With the filter option available right next to the list of products, Christina narrows down the list to a smaller range meeting her needs. Also with the quick option to sort products

by price range she can pick a price that fits her budget.



Christina can compare between babysitting service sellers by checking their rating and buyer's comment on the item's product page.



She can also add several items to the cart so that she can make comparisons before making the final decision.

A hand-drawn sketch of a shopping cart interface. At the top left is a simple drawing of a shopping cart. Below it, a list of items is shown with three columns: an index number, a price with a dollar sign, and a name. The first two rows are numbered 1 and 2. Below these are three vertical ellipses indicating more items. In the bottom right corner, there is a rectangular button labeled "Check out".

	\$	Price	Name
1.	\$	Price	Name
2.	\$	Price	Name
⋮	⋮	⋮	⋮

Check out

Emilio

As CsGetDegrees' treat a service as a product, Emilio create a product listing in the create product popup. Emilio writes the details on how the service is provided and specified that he is willing to provide further information that he does not wish to publish publicly if the potential buyer contacts him.

A hand-drawn sketch of a "POP UP / Create Product" form. The form is titled "POP UP / Create Product" at the top. It contains several input fields: "Title:", "Quantity:", "Categories:", "Department:", "Courses:", and "Tags:". To the right of the "Quantity:" field is a checkbox labeled "check if extended". Below these fields is a large "Description:" text area. At the bottom right is a "Price:" field. At the bottom left is a checkbox labeled "Agree to Terms & Agreement". At the bottom right is a "Create" button. A note "Description of the Product" with an arrow points to the "Description:" field.

POP UP / Create Product

Description of the Product

Title:

Quantity: ☐ check if extended

Categories:

Department:

Courses:

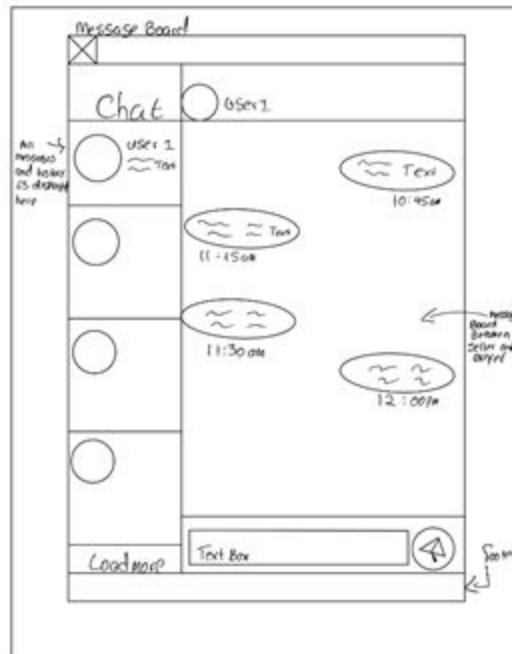
Tags:

Description:

Price:

☐ Agree to Terms & Agreement

Emillio receives the message from the potential buyers in a private chat room. Therefore, he communicates with the buyers.



This messaging feature allows Emilio to easily make arrangements for scheduling his business with customers.

III. High level Architecture, Database Organization

When the user is in the process of posting a product or service, there will be a field where they can put a link to an image. This link will be taken as a string, the string will be stored in the database, in the Image Table.

The search bar takes a string from the user. The name column in the Product table of the database is search for existing product name strings that contain the user input string as a keyword. The result is a Result table which contains the relevant products.

(Higher level description):

The User inputs a string in the search bar and this string is used to make a query to the MySQL database, which performs Linear Searches on the tables in order to return the correct products, the classes they are associated with, and the departments those classes are in. In other words, the search and filter implementation is represented in our system by querying the MySQL database and letting MySQL filter the results for us so we can pass them straight into our app.

Our database will have an endpoint for getting chats. This will allow us to query the Message table in the database for other users and messages someone has initiated a chat with.

Our database will have an endpoint for getting images uploaded by users. This will allow us to query the Image table in the database for images associated with a specific product.

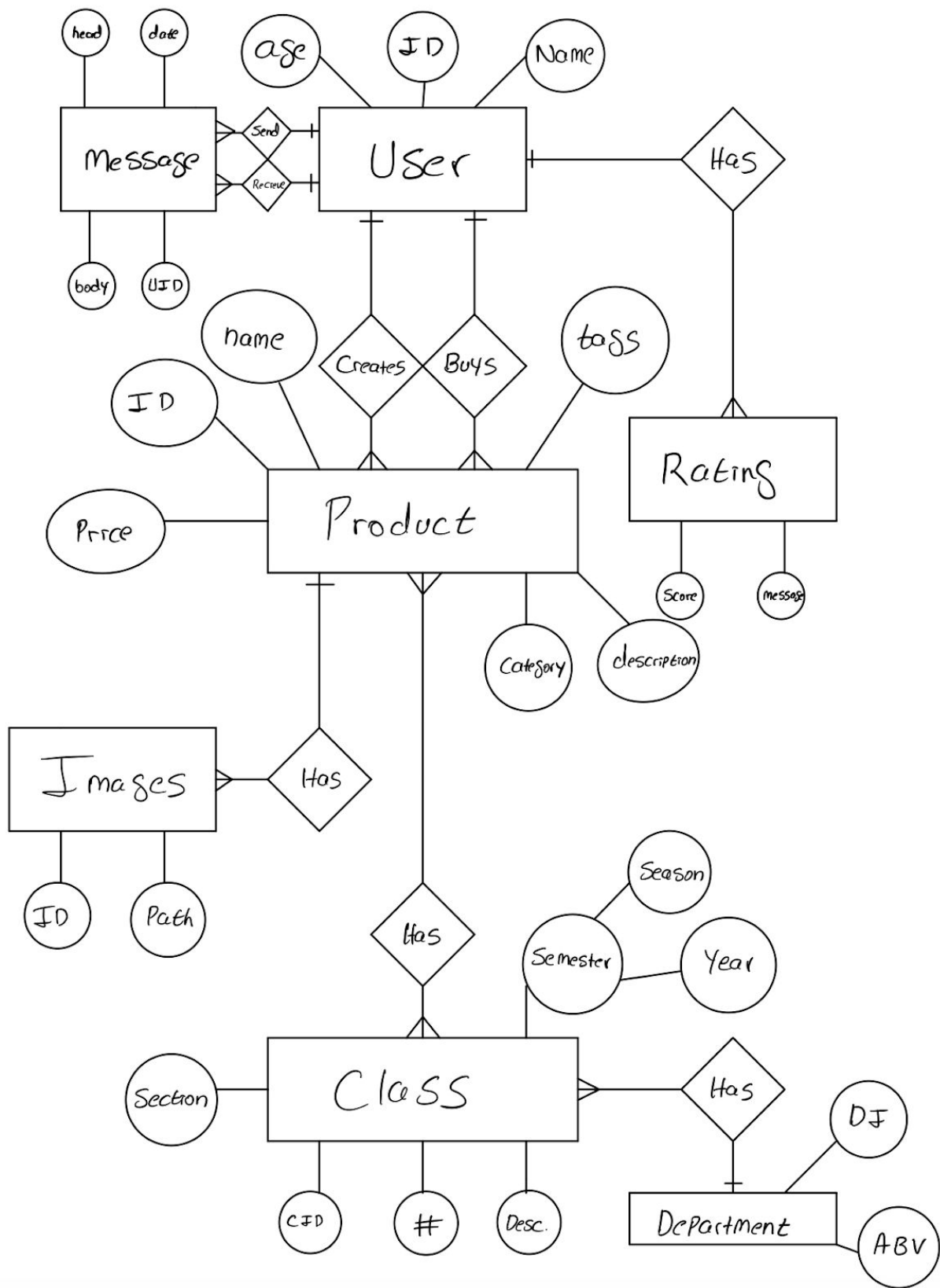
Our database will have an endpoint for getting products posted on the site. This will allow us to query the Product table for products that match a specific search term and use the product_ids of those rows to query the Class and Department tables respectively. Products belong to classes and classes belong to departments. It is important to get this information from the database because the way to navigate to a product is through a menu of departments, which presents a stealth slide-out menu of classes, which can each be selected to view the products. The UI described here can be seen visually in the mockups above in section 2.

Our database will have an endpoint for getting ratings posted for users on the site. This will allow us to query the Rating table for ratings that match the searched user_id.

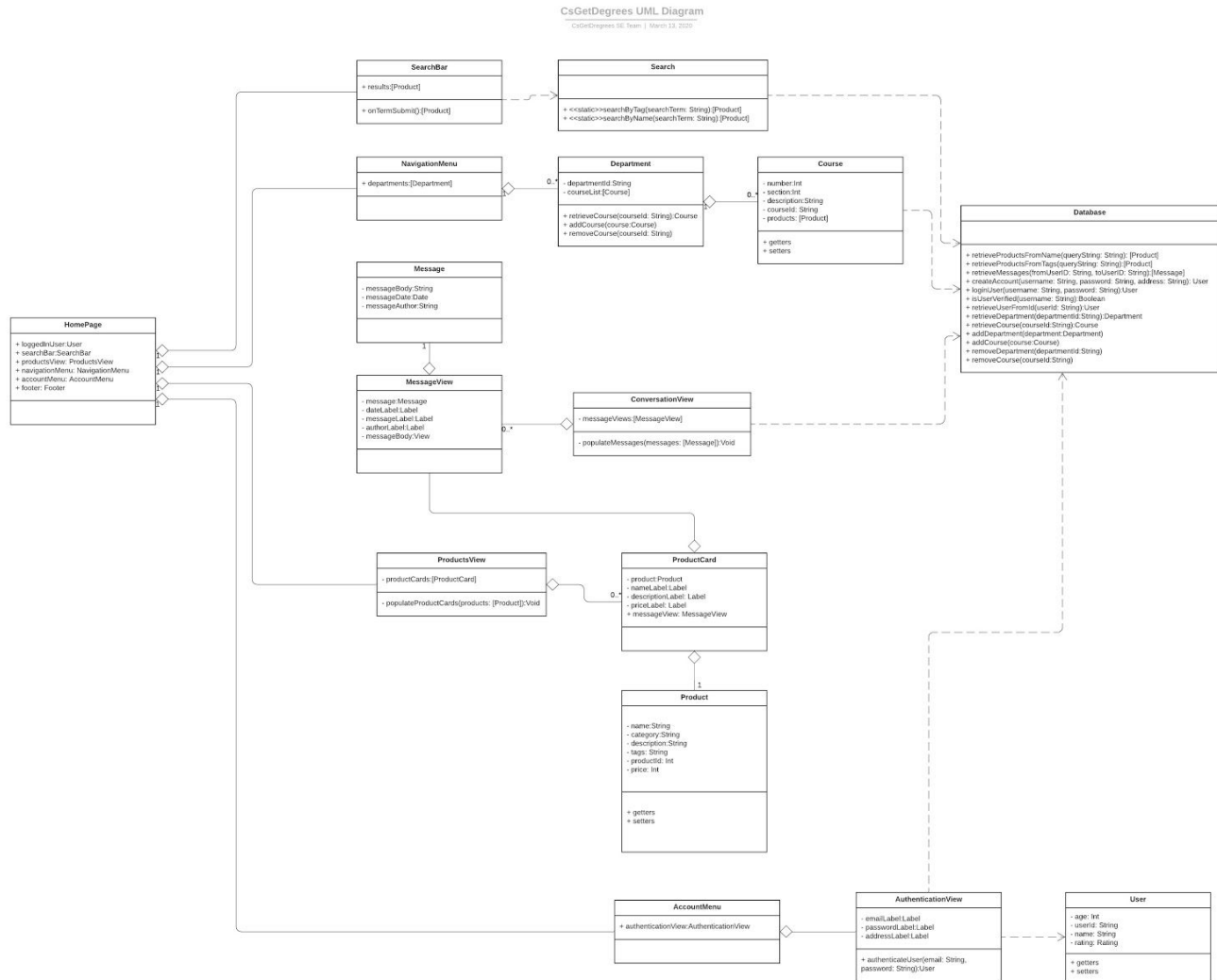
Our app will include a rating system. This will make it very easy to sort listed products by quality. After the transaction is complete, the buyer will be able to rate the seller. By only allowing users to review other users once a transaction is complete, we can ensure that nobody is getting unfairly reviewed. All users, however, can see the rating of a user selling an item.

Our app will include a messaging system. Only buyers can initiate a conversation. We do this so that sellers can't mass spam users to buy a product and to promote messages being used only when a buyer is interested in a product. Messaging can be initiated between users via a sellers' listed product page.

Our app will include a transaction system. We don't store credit cards or connect to a payment api. Instead, we are implementing a fake currency designed to simulate what it would look like *if* the app were connected to a payment api. When signed in, users are allotted a fixed amount of this currency. When the "Buy" button is pressed on a product's listing page, a confirmation screen is shown. Once the order is confirmed, the currency is transferred between users.



IV. High Level UML Diagrams



V. Key risks for our project at this time

Skill risks

Most of our team are new to the programming language or script (Javascript and Python) and frameworks we are using in our project. We do not really know where to start.

We will have a group tutorial on Python so that we can discuss everything we find confusing as a group. At the same time, we can understand each other's progress.

We have split the team into front-end and back-end groups according to the skills that each respective member feels confident with.

Schedule risks

Due to problems in commuting, different class and work schedule, we are not able to meet each other often. This might cause lack of communication between our team and ultimately leads to misunderstanding.

We are not sure of the flexibility of CEO/Professor in terms of scheduling. We are also unsure of how long each milestone will take and are therefore, not sure of how much time we'll want to allot to each milestone if there is flexibility.

Technical risks

The response time of the server takes too long. It might be very bad for the user experience as the user most probably wants to enjoy a lag free environment.

We should optimize the performance of the server. Besides that, we should seek ways to increase the bandwidth of the server.

VI. Project management

Our main source of communication on Milestone two included Google docs, Discord, and Slack. As we progressed through our planning stage, we slowly incorporated other tools to help design our ideas. For example, one such tool that we used to draw out our ideas is called LucidChart. LucidChart became a valuable tool in helping us design and integrate our uml charts together between the backend team and frontend team. There are other tools we are beginning to explore, one such tool that we could perhaps use for future task planning is Trello. Trello can help us manage how we communicate and how we see the bigger picture of our final product. Tools similar to Trello can be a great advantage when we are completing tasks, due to its ease of use and ease of understanding. Overall as we grow and learn, we'll begin to incorporate different ideas and different tools into our project, as a fast learning team, any and all tools can be easily integrated into our project management.