

# **Final Project for SW Engineering Class**

## **CSC 648-848 Fall 2018**

### **Team 03**

### ***EduGator***

December 22, 2018

#### **Team Members**

James Andrews [jandrew3@mail.sfsu.edu](mailto:jandrew3@mail.sfsu.edu) (Team Leader)

Ian Dennis

Ivan Varela

Chris Johansen

Liyao Jiang

Shirin Namiranian

Kenneth Surban

**Demo URL:** <https://team03-648.herokuapp.com/>

# Product summary:

**Product Name:** EduGator

## **Major Committed Functions**

- Login/Registration
- Search (with pull down categories, text entry)
- Search Results
- Item details
- Seller dashboard (shows selling items and messages)
- Admin
- Messaging (buyer to seller only)
- Tutoring Services

**Our Product:** We provide a marketplace for SFSU students to buy and sell general items, tutoring services, and coursebooks.

**Competitive Edge:** In addition to facilitating a marketplace for SFSU students to buy and sell items, our product also allows for SFSU students to seek and offer tutoring services from their peers, and takes advantage of the SFSU niche by mapping textbooks for courses to the appropriate course.

**Product URL:** <https://team03-648.herokuapp.com/>

## Milestone Documents

# SW Engineering CSC648/848 Fall 2018

## *EduGator* Team 03

Milestone 1  
October 4th, 2018

### Team Members

James Andrews      [jandrew3@mail.sfsu.edu](mailto:jandrew3@mail.sfsu.edu) (Team Leader)  
Ian Dennis  
Ivan Varela  
Chris Johansen  
Liyao Jiang  
Shirin Namiranian  
Kenneth Surban

Revision Number	Description	Date
1.0	Initial Version	10/04/2018
2.0	Revised based on comments	10/10/2018

### 1. Executive summary

As a student, the process of buying and selling books, old room furniture, or even getting tutoring can be a hassle. You can check Amazon or eBay for your textbook, but can't be sure if you are getting the right volume you need for the class, or if it's the best price. EduGator aims to make this process of buying and selling items more streamlined for the SFSU campus. Since the application will be tailored to SFSU students, we will have books sortable by SFSU course they pertain to, and also sort other items into categories that students would find useful. Additionally, we plan to make the process of finding a tutor as easy as possible by allowing SFSU community members themselves create listings to tutor their fellow students in courses offered by the university.

EduGator aims to allow students to buy and sell items such as electronics and furniture, but our main focus will be on allowing students to find books relevant to SFSU courses and allow students who need additional help to find tutors within our community. Books will have attributes that link to a specific course, and our tutoring service will allow users to sell their services for that course.

We chose to focus mainly on books and the tutoring service because we feel that is a critical piece missing in the marketplace for SFSU students. We will also provide the option to sell and buy other items, so users won't be limited to just books and tutoring, and more importantly, so that our product can grow to become the new hub for the SFSU student marketplace.

Our product simplifies the process of buying and selling items to SFSU students. When it comes to books, users will have identifiers that connect them to specific SFSU courses, which facilitates SFSU students finding books for their classes. These identifiers will also be used for students looking for and selling tutoring services.

EduGator has an innovative focus on selling books and tutoring services. The tutoring will be specific to the SFSU campus, allowing our product to hit a niche market that has yet to be serviced, and allows us to serve SFSU student community as closely as possible.

Competitor apps include Amazon and Ebay, who focus broadly on selling all types of items, and do not specifically target SFSU community needs. We plan to tailor our service to align with the needs of a SFSU student.

Our team consists of seven computer science students from SFSU that want to simplify the process of selling items and buying items for our fellow students, and approve the overall quality of campus life here at SFSU.

## 2. Personas and Use Cases

### 2.1 About Elisa - “I love to explore and try new things”

Elisa is a freshman business major student in SFSU. She is new to the city, lives in an apartment close to school and wants to settle down quickly to college life. She likes shopping online and social media a lot. She is a fast-paced, socialized person, and makes friends easily. Obtain everything for her new apartment and school materials at a reasonable price and quickly. Likes to share good deals she finds online with her friends.



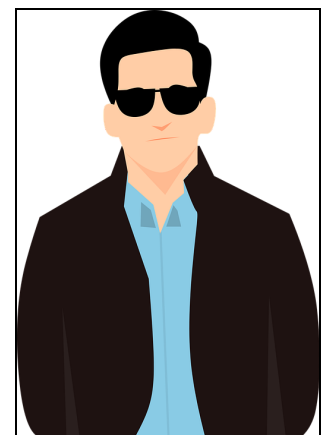
### 2.2 About Mike - “I love technology”

Mike is a busy SFSU student who is majoring in computer science and is proficient with technology. His phone is with him 24/7 and he never fails to respond to his emails in a timely manner, especially since he spends most of his time on his laptop. Mike is a hard worker and loves jobs where he can work remotely. Mike wants to do a great job as an admin of EduGator so that he can be recommend for other admin jobs in the future. He wants to have power over postings because he hates other websites that have unhelpful and useless postings.



### 2.3 About Tyler - “I love nice things and money”

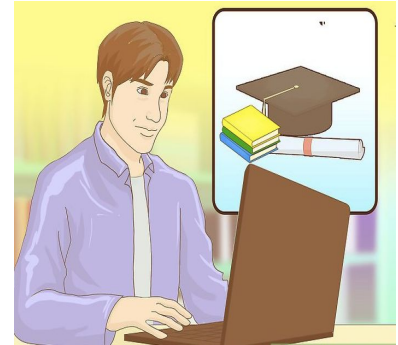
Tyler is from Los Angeles, CA. He is a 23 year old student studying Accounting at San Francisco State University. He is in his junior year and has been at SFSU for 5 years. He lives in an off-campus apartment and is financially dependent on his upper middle class parents. Tyler has a bad addiction at spending his monthly allowance for school. He will browse the Internet, look for good deals, and make spontaneous purchases. Tyler is really in to being always up to date with his technology and loves getting the latest Apple product. Tyler often finds himself having a lot of tech, items, and old books he wants to get rid of but wants some money out of it. Tyler wants a platform where he can sell his mountain of



unused belongings and old books to offset his bad spending habits and make extra money.

#### **2.4 About Arnold - “I love having enough money for rent”**

Arnold is a 21 year old student who is a senior at SFSU and is majoring in computer science. He loves computer science, math, and physics; he has such a passion for those topics that he enjoys teaching and tutoring students in those topics. Ever since moving to San Francisco Arnold worries about not having enough money to pay his rent, he’s always on the lookout for ways to make money. He wants to be able to easily make money while he is studying and needs a job that doesn’t require much travel to be flexible with his class schedule.



#### **Use Case 1 - Unregistered user browsing items and buying items**

On a Saturday morning, Elisa is browsing EduGator, a sell and buy website for SFSU students which a friend told her about. She wants to buy a textbook for her business class, a brand new one is very expensive for her so she wants to find a used textbook. She is just checking out the site without registering first. She searches for textbooks by typing in her class number into the search bar, and surprisingly she finds the class book that is sold by a former student at an incredibly low price. She tries to contact the seller and is prompted to register or login. She creates her account, leaves a message to the seller, and the seller contacts her and they make a deal the same day.

#### **Use Case 2 - Admin managing posts and banning users**

Mike is an admin for EduGator. As an admin Mike can’t buy or sell items or services, his only responsibility is to administer EduGator. When administering EduGator Mike is responsible for the status of items and services posted. When an item or service is posted by a seller Mike receives a notification at his e-mail address that an item or service is waiting to be reviewed. That is to say that each item or service are bound to a flag. When an item or service is posted the default value of the flag is pending. When a flag is pending the item or service is not visible to potential buyers. For the flag of an item or service to be set to live Mike must review the item or service and approve it. If the item or service are not up to Mike’s standards he can contact the seller and ask them to change the posting, or if the posting is inappropriate Mike can delete the posting and ban the user from creating further postings. If by some chance Mike does

approve an inappropriate posting, users can flag the posting as inappropriate, in which case Mike will receive a notification at his e-mail address, Mike can then delete the posting.

### **Use Case 3 - Seller checking the status of his posts and messages from buyers**

Tyler finds a buy and sell site exclusively for SFSU students. Without registering, he is able to browse through the site. He sees the site is free and easy to use. Tyler wants to make a post for the items he wants to sell. He plans to sell his Apple Series 1 Watch and some old textbooks. He is prompted with a registration login screen. He signs up and creates a post, including the name of the item, a photo, a description, the category, and the condition. For the books, he includes the same: name, photo, description, category, condition, and also the book's course title. After some time, he wants to check the status of his post. He goes to our site, logs in, and sees the condition of his post. He checks back the next day and sees on a dashboard someone messaged him about his book.

### **Use Case 4 - Seller posting items**

For Arnold to pay his rent and tuition, he needs to make money while he is at school. He needs a job with flexible hours to work and have enough time to study. Arnold is feeling a little worried. Thus he has to offer some services to other students for making money. He posts on EduGator in the tutoring section, he creates posts in the Math, Physics, and Computer Science sections. Arnold can choose how much he charges each student per hour for his tutoring services. After his post has been up for a couple of hours EduGator notifies him that he got a request from Suzy who is a junior at SFSU and majoring in computer science, to tutor her about the Java language for her CS413 class. Since Arnold got connected with Suzy through the website, he is able to help her for an hour. He feels safe and secure that Suzy will pay him money for his time helping her with Java. Arnold can sit down and enjoy the rest of his evening by checking his bank account and not worrying about how to pay his rent and bills.



## **3. Data Description**

### **3.1 Unregistered Users:**

Can browse items, course books, and tutoring services. Does not need to register or login to do this.

### **3.2 Registered Users:**

Can browse, post, view the approval status of, remove, and send messages to other users about items, course books, and tutoring services. They need to be registered and logged in to access this functionality for their account. Each account is also linked to a user registration record that the registered user completed when initially created their account.

### **3.3 Admin:**

An employee of the company hired to change the approval status of or remove a course book, item, or tutor service that a registered user has posted. Needs to be logged in.

### **3.4 Item:**

An item available to be purchased, other than a course book. Has a name photo, price, description, category, condition, approval status, and is associated with the registered user who posted it.

### **3.5 Course Book:**

A book associated with a particular course at SFSU. Has a name, photo, price, description, course, condition, approval status, and is associated with the user who posted it.

### **3.6 Tutor Service:**

A post by a registered user to tutor for a specific course. It has a price, description, course, and approval status.

### **3.7 Approval Status:**

The approval status of a particular tutor service, item, or course book. Viewable by the admin and the registered user who posted the item for sale. Only the admin can change this value.

**3.8 Course:**

The name of a course at SFSU. Each course book and tutoring service has a course associated with.

**3.9 Category:**

A classification for items which enables them to be sorted into different groups.

**3.10 Message:**

A message sent between two registered users. It is used for registered users to communicate about tutoring services, course books, and items.

**3.11 User Registration Record**

A record that an unregistered user completes to become a registered user on the site. The user registration record includes the name of the user, the user's password, and the user's email.

## **4. High-Level Functional Requirements**

**4.1 Search Bar:**

Registered and unregistered users shall be able to search items by name, category, and condition. Registered and unregistered users shall be able to search books by name, course, based on the type of item being searched. Registered and unregistered users shall be able to search by course for tutors.

Admins shall be able to view items, books, and tutoring service posts that have a pending status.

**4.2 Filtering types:**

Unregistered, registered, and admin users shall be able to filter by type of item, price, and category.

**4.3 Item filters:**

Registered, unregistered, and admin users shall be able to filter type of item by: furniture, books, general items, etc.

**4.4 Tutoring:**

Registered, unregistered, and admin users shall see tutors related to the book, professor, or class number they are searching.

#### **4.5 Browsing:**

Registered and unregistered users shall see a list of items/services to browse that other users have posted with approved status.

Admin users shall see a list of items/services to browse that have not yet been approved(pending status).

#### **4.6 Posting Items/services:**

Registered users shall be able to post their personal items to be sold. Registered users shall be able to set the price, condition, type, and name of the item. Posted items and tutoring services shall be subject to approval via administrators of the site.

#### **4.7 Messaging for Buying Items/Services:**

Registered users shall be able to contact other users to buy items/services they have posted.

#### **4.8 Admin controls:**

Admins shall be the only users to approve posts. Admins shall be able to remove posts.

## **5. Non-Functional Requirements**

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.
3. Selected application functions must render well on mobile devices
4. Data shall be stored in the team's chosen database technology on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time

6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English.
8. Application shall be very easy to use and intuitive.
9. Google analytics shall be added
10. No e-mail clients shall be allowed
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
12. Site security: basic best practices shall be applied (as covered in the class)
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
14. The website shall prominently display the following exact text on all pages *"SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

## 6. Competitive Analysis

	Ebay	OfferUp	EduGator
Demographic Emphasized Services	+	++	++
Tutoring	-	-	++
Browse	++	++	+
Filtering	++	+	++
Text Search	++	+	+

Our future product's competitive advantage lies in its tailored service to the SFSU campus community. We offer a tutoring marketplace for SFSU students, which is a unique facet of our product as is evidenced in the competitive analysis chart. While we may not have as strong browsing, searching, and filtering capabilities as our competitors, we will focus on our niche target demographic of SFSU students, while still delivering the capability to browse, filter, and search through our content.

## 7. High-Level System Architecture

**Server Host:** Heroku 1vCPU 512 MB RAM

**Operating System:** Ubuntu 18.04 Server

**Database:** PostgreSQL 10.1

**Web Server:** Node 8.0.0

**Server-Side Language:** Nodejs

**Additional Technologies:**

**Web Framework:** Expressjs

**Text Editor:** Atom

**Web Analytics:** Google Analytics

**ReactJs:** 16.5

**jQuery:** 3.3.1

**pg-promise:** 8.4.6

**Styled-components:** 3.4.9

**Bootstrap:** 4.1.3

## 8. Team

- James Andrews - Team Lead
- Ivan Varela - Front-End Lead
- Shirin Namiranian- Front-End Developer
- Kenneth Surban - Front-End Developer
- Ian Dennis - Back-End Lead
- Liyao Jiang - Back-End Developer
- Chris Johansen- Back-End Developer

## 9. Checklist

- Team found a time slot to meet outside of the class - **DONE**
- Github master chosen - **DONE**
- Team decided and agreed together on using the listed SW tools and deployment server - **DONE**
- Team ready and able to use the chosen back and front end frameworks and those who need to learn and working on it - **DONE**
- Team lead ensured that all team members read the final M1 and agree/understand it before submission - **DONE**

# SW Engineering CSC648/848 Fall 2018

## *EduGator* Team 03

Milestone 2  
October 18th, 2018

### Team Members

James Andrews      [jandrew3@mail.sfsu.edu](mailto:jandrew3@mail.sfsu.edu) (Team Leader)  
Ian Dennis  
Ivan Varela  
Chris Johansen  
Liyao Jiang  
Shirin Namiranian  
Kenneth Surban

Revision Number	Description	Date
1.0	Initial Version	10/18/2018
2.0	Revision of CEO comments	10/31/2018

# **1. Data Description v2**

## **1. Unregistered Users:**

Shall be able to browse items, course books, and tutoring services without having to log in or register, but cannot post the aforementioned, and cannot message registered users about them.

## **2. Registered Users:**

Shall be able to browse, post, view the approval status of, remove, and send messages to other users about items, course books, and tutoring services. They must be registered and logged in to access this functionality for their account, and each account shall be linked to a user registration record that the registered user completed when initially creating their account.

## **3. Admin:**

An employee of the company who shall be able to change the approval status of or remove a course book, item, or tutor service that a registered user has posted. They shall not register for an account, but must be logged in to gain access to the aforementioned functionality. They shall not message registered users about or post items, course books, or tutoring services.

## **4. Post**

An item, course book, or tutor service that a registered user can add to our website. They shall be approved by the admin before being searchable/viewable.

## **5. Item:**

An item available to be purchased, other than a course book. It shall have a timestamp, name, photo, price, description, category, condition, approval status, post ID, and an associated registered user (the one who posted it).

## **6. Course Book:**

A book associated with a particular course at SFSU. It shall have a timestamp, name, photo, price, description, course, condition, approval status, post ID, and an associated user (the who posted it).



**7. Tutor Service:**

A post by a registered user to tutor for a specific course. It shall have a timestamp, name, price, description, course, approval status, post ID, and an associated registered user (the one who posted it).

**8. Approval Status:**

The approval status of a particular tutor service, item, or course book. It shall be viewable by admins and by the registered user who created the post, but only admins shall be able to change this status. The status shall be either pending or approved.

**9. Course:**

The name of a course at SFSU. Each course book and tutoring service shall have a course associated with, and the courses shall be pre-populated in the database to avoid user error.

**10. Category:**

A classification for items which enables them to be sorted into different groups that the user shall select when posting an item for sale. The categories shall be pre-populated into the database to avoid user error.

**11. Message:**

A message sent between two registered users that shall have a maximum length of 256 characters, and a message ID. It shall be used for registered users to communicate about tutoring services, course books, and items.

**12. User Registration Record**

A record that an unregistered user shall complete to become a registered user on EduGator. The user registration record shall have the name of the user, the user's password, and the user's email.

**13. Condition**

The state of an item or course book. This selectable conditions that a user can choose from shall be pre-populated into the database, and shall be either "excellent", "good", "fair", or "poor".

**14. Price**

The cost for a tutoring service, course book, or item. It shall be set by a registered user and shall be a floating point value with two decimal places.

## 2. Functional Requirements v2

### Priority 1:

#### 2.3 Unregistered User:

##### Browsing:

Unregistered users shall see a list of items, tutoring services, and course books to browse that other registered users have posted and have been approved by an admin.

##### Search Bar:

Unregistered Users shall be able to search for approved posts using a free text search on the description and name of a post.

##### Post Filters:

Unregistered users shall be able to filter by type of post: item, course book, or tutoring service.

Unregistered users shall be able to filter items by: category, price, condition.

Unregistered users shall be able to filter course books by: course, price, condition.

Unregistered users shall be able to filter tutoring services by: course and price.

##### User Registration:

Unregistered users shall be able to register for an account on EduGator.

#### 2.4 Registered User:

##### Posting Items/services:

Registered users shall be able to post their personal items to be sold. Registered users shall be able to set the price, condition, type, and name of the item.

##### Browsing:

Registered users shall see a list of items and tutoring services to browse that other users have posted with approved status.

##### Messaging for Buying Items/Services:

Registered users shall be able to contact other users to buy items/services they have posted.

#### Search Bar:

Registered users shall be able to search items by name, category, and condition.

Registered users shall be able to search books by name, course, based on the type of item being searched.

Registered users shall be able to search by course for tutors.

#### Post Filters:

Registered users shall be able to filter by type of post: item, course book, or tutoring service.

Registered users shall be able to filter items by: category, price, condition.

Registered users shall be able to filter course books by: course, price, condition.

Registered users shall be able to filter tutoring services by: course and price.

#### User Registration:

Registered users shall be able to login into their account on EduGator.

### **2.5 Admin:**

#### Posting Items/services:

Posted items and tutoring services shall be subject to approval via admins of the site.

#### Browsing:

Admin users shall see a list of items/services to browse that have not yet been approved(pending status).

#### Search Bar:

Admins shall be able to view items, books, and tutoring service posts that have a pending status.

#### Post Filter:

Admins shall be able to filter posts by approval status.

#### User Registration:

Admins shall be to login to their account on EduGator.

## Priority 2:

### 2.5 Admin:

#### Admin controls:

Admins shall be the only users to approve posts.

Admins shall be able to remove posts.

#### Deleting Post:

Admin shall be able to delete a post.

## Priority 3:

### 2.9 Registered User:

#### Wanted Items:

Register users shall be able to post items that they are currently looking for with a negotiable price amount.

#### Reporting User:

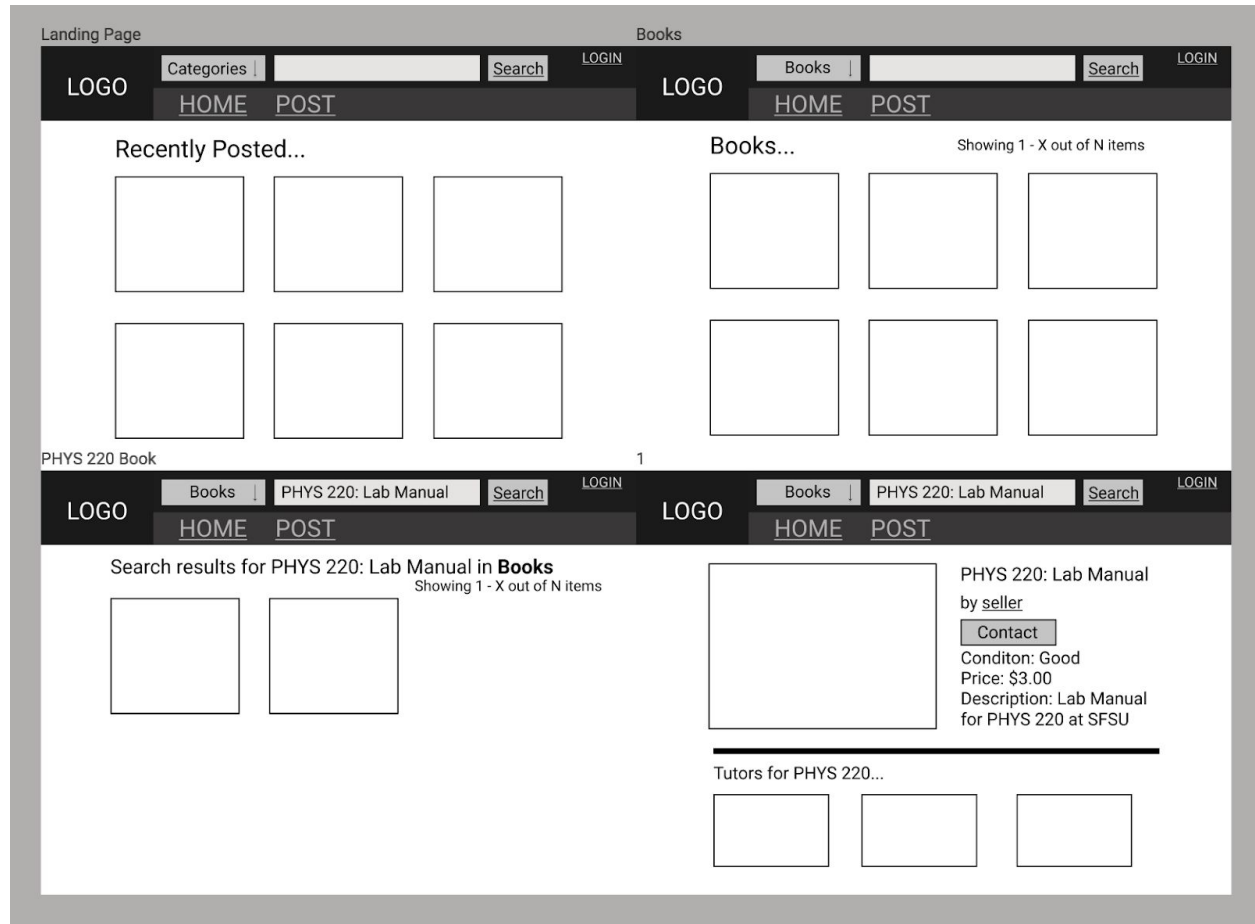
Registered users shall be able to report other users.

#### Reporting posts:

Registered Users shall be able to report posts of other users that they believe are offensive or nondescript.

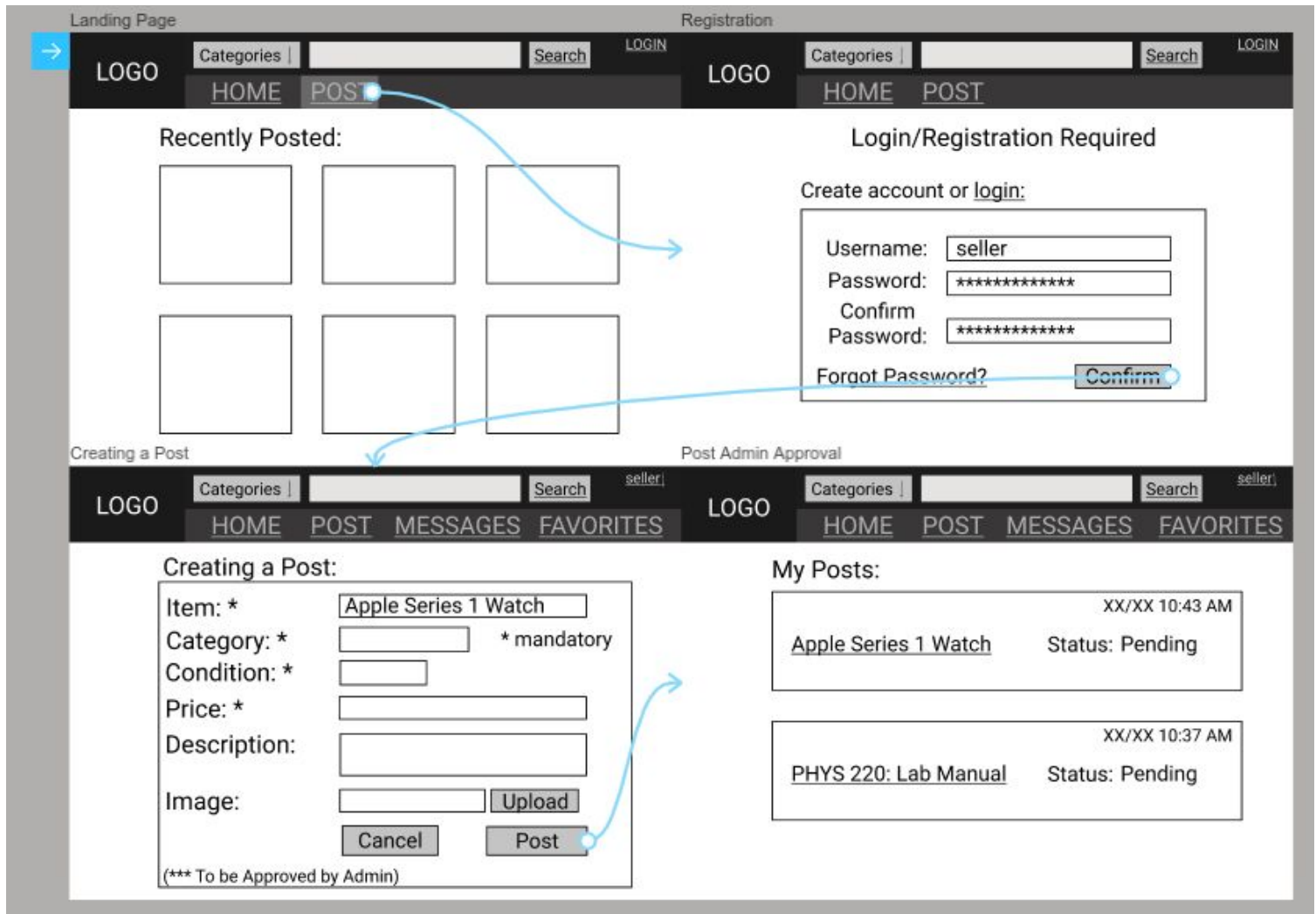
### 3. UI Mockups and Storyboards

#### 3.1 Use Case 1 - Unregistered user browsing items and buying items



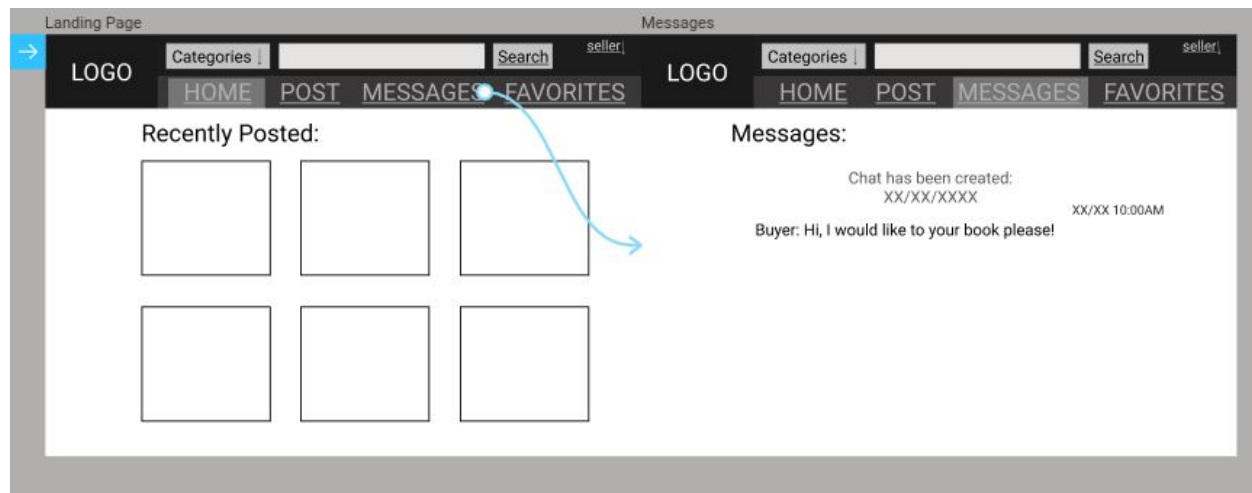
User starts at Home page and then clicks on the categories drop down and selects the books categories, the page then changes to show available books. The user specifies what they want by using the search bar, items matching the search query are then displayed. The user then clicks on an item that interests them and is greeted by a page containing all the information about the post.

### 3.2 Use Case 2 - Seller wants to Create a Post and Check the Status of the Posted Item



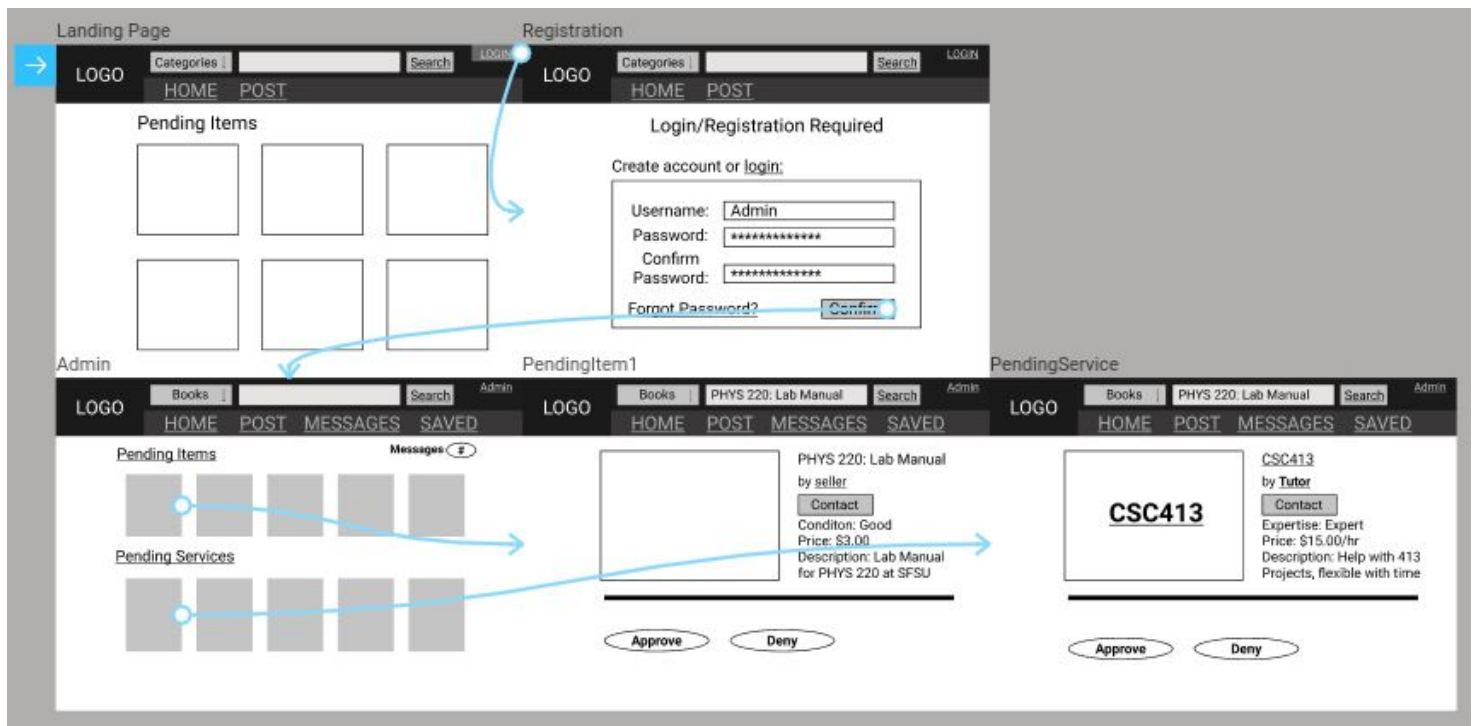
User starts at the landing page and wants to make a post. The user clicks on the Post tab found on the navigation bar and is prompted with a mandatory login/registration page. Creating an account the user is redirected to a form page where the user can fill out fields for the item he/she is selling, such as the name of the item, category, price, and if he or she wants they can add a description and a photo. Clicking post will submit all the data the seller enters in the form field, and then the seller is routed to a status page of the Sell's posts. This page will give the status of the user's posted items, and will change when admin reviews the post.

### 3.3 Use Case 3 - Seller wants to Check Messages



When a seller is logged in, the seller can check if anyone messaged by clicking on the Messages tab which can be found on the navigation bar. Clicking on the Messages tab will redirect the seller to a page where he or she can view the number of messages he has received and the messages he gets from interested buyers.

## Use Case 4 - Admin Managing Posts



Admin starts at the landing page. Clicks on the login link and logs in. Logging in, the Admin is redirected to an admin page of all the items and services waiting to be reviewed. Clicking on a pending item or service will redirect them to that post with an admin only option of approve or deny.



## 4. High Level Architecture, Database Organization

### 4.1 DB Organization

#### User Registration Record

- email - varchar (PK)
- username - varchar
- password - varchar

#### Admins

- email - varchar
- username - varchar
- password - varchar
- admin Id - varchar (PK)

#### Items

- post id - varchar (PK)
- email - varchar (FK)
- name - varchar
- category - varchar
- description - varchar
- price - decimal
- condition - varchar (FK)
- photo - varchar
- timestamp - timestamp

#### Conditions

- condition - varchar (PK)
- post id - varchar (FK)

#### Approved Posts

- post id - varchar (PK)
- status - varchar

#### Course Books

- post id - varchar (PK)
- email - varchar (FK)
- name - varchar
- description - varchar
- price - varchar
- condition - varchar (FK)
- course - varchar
- photo - varchar
- timestamp - timestamp

#### Tutor Services

- post id - varchar (PK)
- email - varchar (FK)
- name - varchar
- description - varchar
- price - varchar
- course - varchar (FK)
- photo - varchar
- timestamp - timestamp

#### Messages

- message id - varchar (PK)
- sender - (FK)
- receiver - (FK)
- message - varchar
- timestamp - timestamp

## **4.2 Media Storage**

We are not using file BLOBs, and will be using Amazon S3 to host our images.

## **4.3 Search/Filter Architecture and Implementation**

1. Items
  - a. Filtering based on SQL queries on category, condition, timestamp and price keys.
  - b. Can be searched by free text by using SQL %like
2. Course Books
  - a. Filtering based on SQL queries on course, condition, timestamp and price keys.
  - b. Can be searched by free text by using SQL %like
3. Tutoring Services
  - a. Filtering based on SQL queries on course and price keys.
  - b. Can be searched by free text by using SQL %like

## **4.4 Our APIs**

REST API that uses express.js as our server and postgres as our database.

## **4.5 Non-trivial Algorithms**

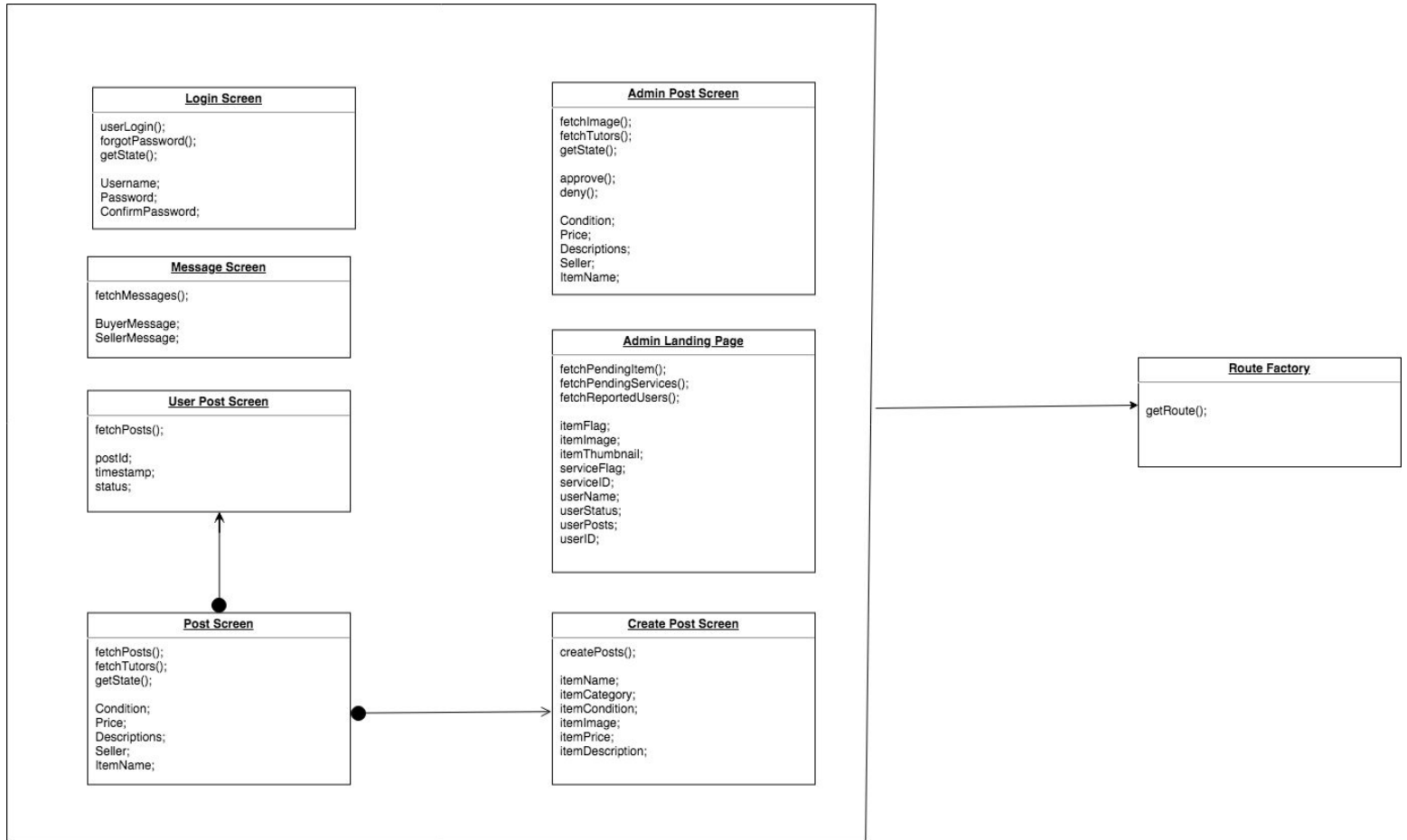
None

## **4.6 Software Tool Changes**

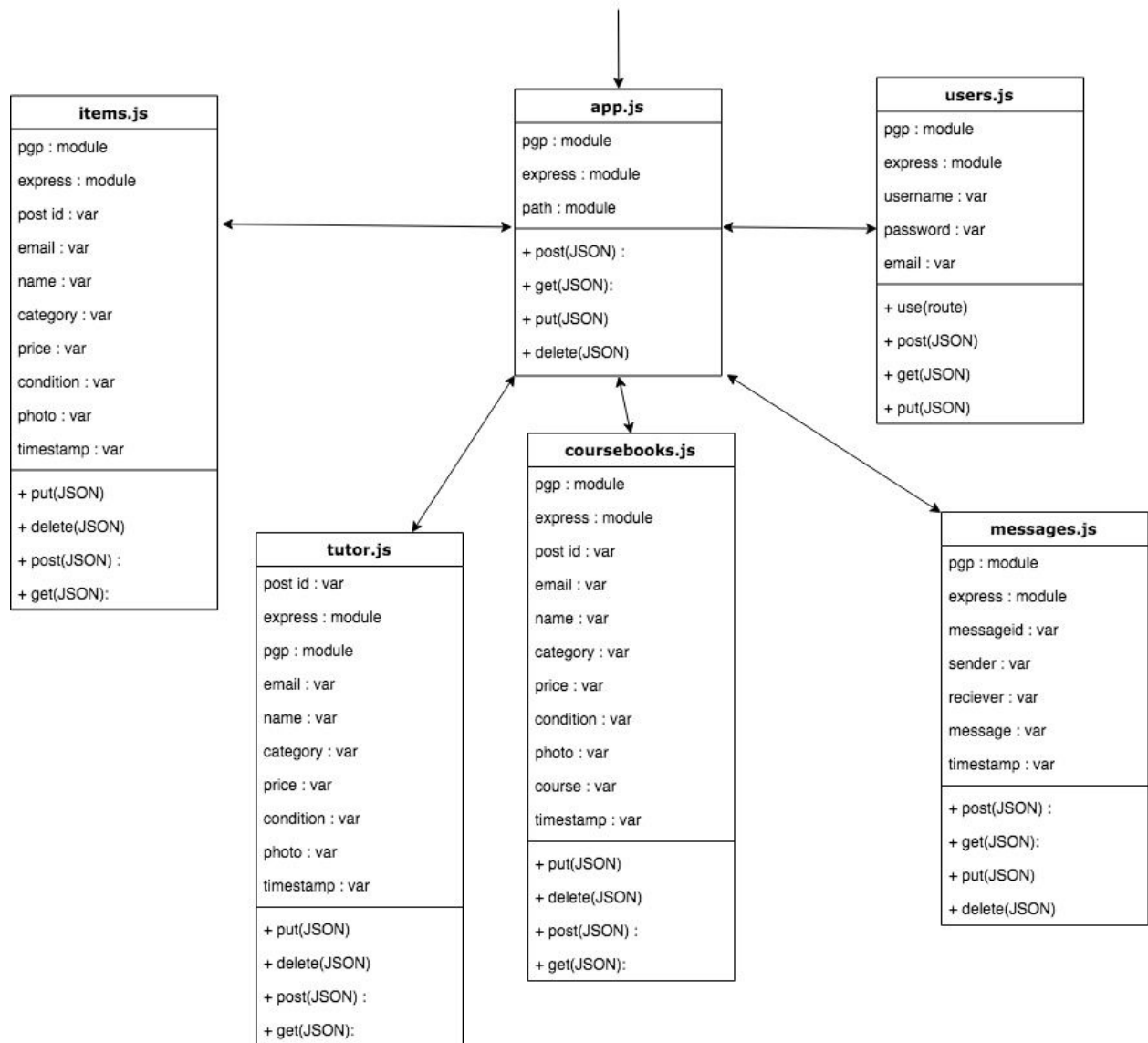
- + Amazon s3 instance to host the images for our website.

## 5. High Level UML Diagrams

### 5.1 Front-end:



## 5.2 Backend:



## 6. Key Risks

### Front-End Risks

#### ***Skills Risk***

The front end team has varying skill levels with CSS, and for most of the team, React and Node are software tools they have not used extensively before. This will cause some time overhead as the team gets familiar with the tools.

To resolve this, the team is studying the tools they are unfamiliar with, and those that are familiar with the tools are helping to get everyone else to a level that they can work productively. The vertical prototype will also act as a vehicle for team members unfamiliar with our tools to see how to use them.

#### ***Schedule Risk***

The team is comprised of students with diverse schedules and unforeseen circumstances could affect the team's ability to deliver on all features.

To compensate for this, the Priority 1 tasks are mainly the bare requirements for the project and our special, competitive feature. If need be, members from the back-end team can assist on front-end work if the back-end completes their Priority 1 tasks before the front-end team.

### Back-End Risk

#### ***Skills Risk***

The most of the backend team has not used the Express framework, Node, or PostgreSQL extensively before this project. This will cause some time overhead as the team gets familiar with the tools.

To resolve this, the team is studying the tools they are unfamiliar with, and assisting others where possible so that the entire team can work productively. The vertical prototype will also allow team members unfamiliar with Express, Node, and PostgreSQL to see demos of syntax and usages.

# SW Engineering CSC648/848 Fall 2018

## *EduGator* Team 03

Milestone 3  
November 16th, 2018

### Team Members

James Andrews      [jandrew3@mail.sfsu.edu](mailto:jandrew3@mail.sfsu.edu) (Team Leader)  
Ian Dennis  
Ivan Varela  
Chris Johansen  
Liyao Jiang  
Shirin Namiranian  
Kenneth Surban

Revision Number	Description	Date
1.0	Initial Version	11/16/2018

***Teamwork: is the team working out, any related issues***

For the most part the team is working out. We were having an issue with one team member on the front end team. The front-end lead and myself sat down with her and outlined exactly what we expected from her and if we don't see any progress we will have to inform you or Anthony.

***Risks: list all actual (not hypothetical) risks (schedule, technical, skills etc.) and plans how to resolve them OIR ask for help.***

- Only real risk we have experienced so far is tasks assigned to Shirin not being completed. The team has met with her, and she seems to be responding to what we have said.

***Coding practices: Have you decided which coding style to use and actually use it? Make sure you sue header comments and some code inline comments too.***

- We are using React for our frontend and Express.js for our backend. Our frontend directory is split by into components, screens, and routes. Components handle styling and basic functionality, screen fetch data from the backend, and routes handle the frontend routing.
- The backend has two directories. Db and endpoints. Db handles all of our database connections (postgres and amazon s3), and endpoints are all of our that the front end can communicate with.

***Usage of proper SE code management practices in github***

1. Do you have master branch that contain only tested code and is carefully managed?
  - We have a master branch which has feature branches and a forked repo in which we connect to Heroku.
2. Do you have development and feature branches
  - Yes our feature branches are branched from master.
3. Are you doing any code review?
  - Yes, before a pull request can be merged with master it requires at least 1 review from an admin.
4. Are your comments on github code commits meaningful to reflect enough details?
  - Where there are comments they are meaningful and describe what is happening in the code.
5. Are all team members committing on github individually
  - Everyone has committed to the project at least one time.

***How did you address site security and safe coding practices:***

- We are using bcrypt to hash and salt our passwords and Passport.js for session management.
- For field validation we are using Bootstraps built-in validation.

***Digital content (e.g. images, video) – have you obtained enough (total of 30 or so) images for e.g. 4-5 product categories for your demo, royalty free***

- As of now we only have 5 of our own photos in the database, but we will be adding more soon.

**Provide URL to home page, running from your server so it can be tested (ensure UI implementation is responsive)**

- <https://team03-648.herokuapp.com/>

**Provide PDF screen shots, ONE PER PAGE, of the following screens (they have to be real coded pages but not necessarily connected to back end)**



## Home page

### Software Engineering class SFSU

Fall 2018 Section 01 Team 03

EduGator

All Categories ▾

Search

Submit

[Home](#)

[About](#)

[Login](#)

[Admin](#)

[Post](#)



#### Skateboard

Catfish BBQ skateboard

**\$80.00**



#### Pen - ballpoint

This pen is pretty good. You should own it. Yaddadimean?

**\$1.00**



#### Wild Animus

This book is about stuff, not sure what, but I bought it. You should buy it, now.

**\$12.00**



## Search Results (NOTE: search parameters chosen by user have to be persistent)

Software Engineering class SFSU

Fall 2018 Section 01 Team 03

EduGator	All Categories ▾	<input type="text" value="book"/>	<input type="button" value="Submit"/>	Home	About	Login	Admin	Post
----------	------------------	-----------------------------------	---------------------------------------	------	-------	-------	-------	------

### Search Results For: book

filtering by: All Categories

Results returned: 1



#### Wild Animus

This book is about stuff, not sure what, but I bought it. You should buy it, now.

**\$12.00**

## Item Posting

Software Engineering class SFSU

Fall 2018 Section 01 Team 03

EduGator

All Categories ▾

book

Submit

[Home](#)

[About](#)

[Login](#)

[Admin](#)

[Post](#)

### Creating a Post:

**Item Name:**

Enter Item Name

**Category:**

Select a Category

**Price:**

Enter Price

**Condition:**

Enter Condition

**Description:**

Enter Description

**Image:**

[Choose File](#) No file chosen

Post

## Admin Dashboard

Software Engineering class SFSU

Fall 2018 Section 01 Team 03

EduGator

All Categories ▾

Search

Submit

Home

About

Login

Admin

Post

## Admin Dashboard

Items

Services

Messages

### Currently pending Items

#	Category	Name	Email	Status	Created	Take Action
4	Book	Wild Animus	yes	pending	23:27:06.454511	<a href="#">Approve</a> <a href="#">Deny</a>
4	Book	Wild Animus	yes	pending	23:27:06.454511	<a href="#">Approve</a> <a href="#">Deny</a>
4	Book	Wild Animus	yes	pending	23:27:06.454511	<a href="#">Approve</a> <a href="#">Deny</a>
4	Book	Wild Animus	yes	pending	23:27:06.454511	<a href="#">Approve</a> <a href="#">Deny</a>

### Currently active Items

**List all major DB tables you have implemented and list their items**

- Admin table
- Approved Post table
- Course Books table
- Items table
- Messages table
- Tutors table
- Users table

**List of requested P1 functions – provided by your CEO**

The following functions constitute minimal P1 list and you must implement and test them for December demo:

- Home page
- Search(with search text validation)
- Search results
- Item details
- Log in and registration
- Item posting
- Message from buyer to seller
- Seller dashboard (for seller items and messages)
- Admin dashboard (backup is to use DB native admin tool but must show ability to approve posted images)

# SW Engineering CSC648/848 Fall 2018

## *EduGator* Team 03

Milestone 4  
December 2, 2018

### Team Members

James Andrews      [jandrew3@mail.sfsu.edu](mailto:jandrew3@mail.sfsu.edu) (Team Leader)  
Ian Dennis  
Ivan Varela  
Chris Johansen  
Liyao Jiang  
Shirin Namiranian  
Kenneth Surban

Revision Number	Description	Date
1.0	Initial Version	12/02/2018
2.0	Revised Version	12/10/2018

# 1. Product Summary:

**Product Name:** EduGator

**Product URL:** <https://team03-648.herokuapp.com/>

**Our Product:** We provide a marketplace for SFSU students to buy and sell general items, tutoring services, and coursebooks.

**Competitive Edge:** In addition to facilitating a marketplace for SFSU students to buy and sell items, our product also allows for SFSU students to seek and offer tutoring services from their peers, and takes advantage of the SFSU niche by mapping textbooks for courses to the appropriate course.

## **Major Committed Functions**

- Login/Registration
- Search (with pull down categories, text entry)
- Search Results
- Item details
- Seller dashboard (shows selling items and messages)
- Admin
- Messaging (buyer to seller only)

## 2. Usability Test Plan

### Test Objectives:

The objective of this test is to see how usable our current item posting interface is. We want to test how quickly a user is able to find the posting button on our home page, how easily they can navigate the posting form as they fill it out, and how understandable our error messages are in the event that a user fills out the form incorrectly.

### Test Plan:

#### A. System Setup:

- Browser: Google Chrome Browser v.71
- Operating System: macOS High Sierra 10.13
- Computer: MacBook Pro Laptop

#### B. Starting Point:

- Home Page of EduGator

#### C. Intended User:

- Any SFSU Student
  - i. Does not need to be tech savvy

#### D. Task: Create a new post for an item, and post it to Edugator.

- Completion Criteria
  - i. The user created a well-formed post that successfully posted to the database.
  - ii. The user completed this task within 2 minutes.

#### E. URL to Test System:

- <https://team03-648.herokuapp.com>



**Questionnaire:**

Question	Strongly Disagree	Mostly Disagree	Neutral	Mostly Agree	Strongly Agree
The format of the site is intuitive and easy to navigate.					
I was able to post an item on the website easily.					
Errors encountered when posting an item were explanatory.					

### 3. QA Test Plan

#### Test objectives:

The objective is to test that the posting form fields for our website function to our specifications. Inputs to the posting form should be checked for validity, and an error message should be shown if the the form field is invalid. The post form shall not allow the user to post an item with an invalid post form.

#### HW and SW setup:

- Hardware
  - MacBook Pro
- Software
  - Web browser (test on 2 latest revisions):
    1. Google Chrome
  - Test Link: <https://team03-648.herokuapp.com/post>

#### Testing Plans:

Number	Description	Test Input	Expected Results	PASS/FAIL
1	Test Empty Category Field	Do not select a category from the "Category" field drop down menu and then press the post button	Form reports an error stating "Category must be selected" in red text	PASS
2	Test Non-Numerical Price	Try to enter a non-numeric price into the "Price \$" field of the post form.	Any non-numeric characters do not appear in the price field	FAIL
3	Test Long Item Name Error	Enter more than 40 characters into the "Item Name" field.	The field becomes red, and an error is added to the form indicating that the input for "Item Name" is too long	FAIL

## 4. Code Review

### Coding Style:

- 2 space indentations
- Variables
  - Camel-Cased
  - Start with a lower-case letter
    - Exception made for module names that are being imported into a file
  - Names shall be clear and meaningful
- Header Comments at the top of each file


### Code Review:

When it comes to code review we use the tools GitHub provides. Each team member works on their own branches and once they feel that their code is ready to be merged into the master branch they create a pull request. Once the pull request has been created other team members will review the code with the final ok being given by the team lead. The following is an example of what our code reviews look like:


### Sample Code Review

`api/endpoints/login.js` Outdated

```
5 + const bcrypt = require("bcrypt")
6 +
7 +
8 + router.get('/registe', (req, res) => {
```

 **idennis7** 21 days ago + 😊 ...

We should probably remove this test route, or change the route name to it.



Resolve conversation

api/endpoints/login.js

Outdated

```
13 +
14 +
15 + router.post('/register', async (req, res) => {
16 +   var saltRounds = 8;
```



idennis7 21 days ago

+ 😊 ...

As a future improvement, we should do some error checking on req.body to make sure that all the parameters were actually passed.



MrApplesnacks 20 days ago

True, we should have that as well as some form verification on the front end



Reply...

Resolve conversation

api/endpoints/login.js

Outdated

```
8 + router.get('/registe', (req, res) => {
9 +   db.any('insert into users(email, username, password) \
10 +     values($1, $2, $3);', ['ema', 'name01', 'password'])
11 +     console.log('this route is working')
```



JamesFTW 21 days ago

Lets remove the console.logs



Reply...

Resolve conversation

api/endpoints/login.js

Outdated

```
17 + var myPlaintextPassword = req.body.password
18 + var salt = bcrypt.genSaltSync(saltRounds);
19 + var hash = await bcrypt.hash(myPlaintextPassword, salt);
20 + db.any("insert into users (email, username, password) values($1, $2, $3
```



idennis7 21 days ago

You don't have to, but I find using ALL CAPS for SQL keywords to be easier to read.



Reply...

Resolve conversation

## 5. Self-Check on Best Practices for Security

### Major Assets:

- Database containing user information and data
- Messages sent between users that could contain personal information
- User postings of items and tutor services
- Metrics on website usage

### Encryption of Password in Database:

Done

### Input Data Validation:

Search bar input only allows for text format and max 40 characters. We are using bootstrap for field validation.

## 6. Adherence to Non-Functional Specs

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).
  - ***DONE***
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.
  - ***ON TRACK***
3. Selected application functions must render well on mobile devices
  - ***ON TRACK***
4. Data shall be stored in the team's chosen database technology on the team's deployment server.
  - ***DONE***


5. No more than 50 concurrent users shall be accessing the application at any time
  - **ON TRACK**
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
  - **ON TRACK**
7. The language used shall be English
  - **DONE**
8. Application shall be very easy to use and intuitive.
  - **ON TRACK**
9. Google analytics shall be added
  - **ON TRACK**
10. No e-mail clients shall be allowed
  - **DONE**
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
  - **DONE**
12. Site security: basic best practices shall be applied (as covered in the class)
  - **DONE**
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
  - **DONE**
14. The website shall prominently display the following exact text on all pages *"SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).
  - **ON TRACK**

End Milestone Documents

# Product Screen Shots

Home Page

SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only  
Section 01 Team 03



All Categories ▾

Post

My Account

Admin

About

Login

## EduGator: Buy & Sell With Peers

Sort By ▾

Viewing: All Categories



\$12.55

### iPhone Charger

An Iphone charger with a lightning cable. Pretty much fresh out of the box.



\$15.45

### IClicker 2

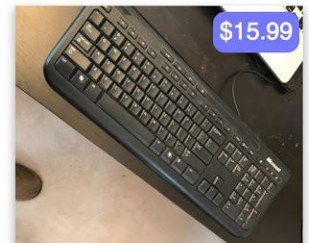
An IClicker you can use for your lectures. I only used this for one class, and I'll even include the batteries.



\$35.75

### Calculator

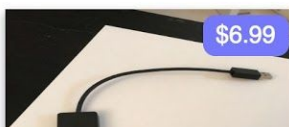
A TI-84 calculator that got me through all of my math classes. It's still in really good condition.



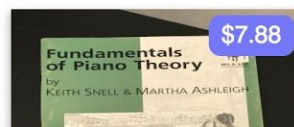
\$15.99

### Keyboard

USB keyboard that functions really well. Just a little bit dusty.



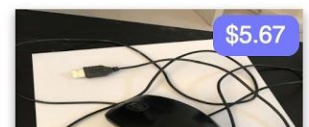
\$6.99



\$7.88



\$9.99



\$5.67



## Item Selection

### iPhone Charger

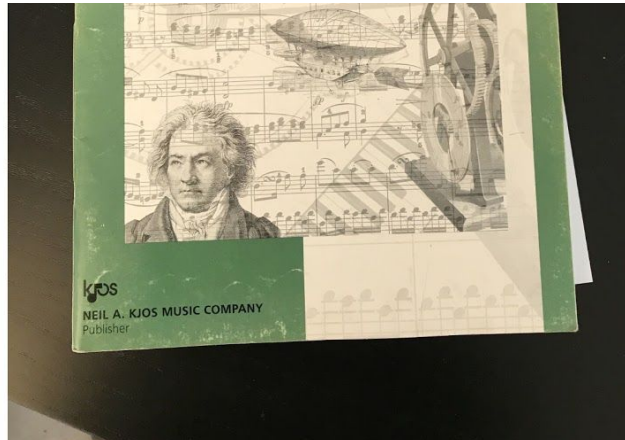
An Iphone charger with a lightning cable. Pretty much fresh out of the box.

\$6.99

### USB Hub

Ever have the problem that your laptop doesn't have enough usb ports? Now you won't.

\$39.95



- **Price:** \$7.88
- **Created At:** 22:17:23.10269
- **Condition:** Lightly Used
- **Description:** A piano theory book for those of you looking to up your theory game. Become the next Mozart!

### Leave a Message:

Send

CLOSE

### Keyboard

USB keyboard that functions really well. Just a little bit dusty.

\$5.67

### Computer Mouse

A well used usb mouse. Still fully functional!

\$7.77

# Registration Page

SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only

Section 01 Team 03



All Categories ▾

Search EduGator

[Post](#)

[My Account](#)

[Admin](#)

[About](#)

[Login](#)

**\* Email**

**\* Password**

**\* Retype Password**

☐ I agree to [Term and Conditions](#)

Register

## User Dashboard

SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only  
Section 01 Team 03



All Categories ▾

[Post](#)[My Account](#)[Admin](#)[About](#)[Logout](#)

## User Dashboard

[Posts](#)[Messages](#)[Messages \(Sent\)](#)

### My Posts

#	Name	Created	Status
36	Binder	23:42:56.2184	active

<https://team03-648.herokuapp.com/myaccount#>

## Admin Dashboard

SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only  
Section 01 Team 03



All Categories ▾

[Post](#) [My Account](#) [Admin](#) [About](#) [Logout](#)

## Admin Dashboard

Items

Currently pending items

Currently active items

#	Category	Name	Email	Status	Created	Take Action
3	Electronics	Iphone Charger	user@gator.com	active	22:10:05.101475	none necessary
2	School Supplies	IClicker 2	user@gator.com	active	22:08:04.158226	none necessary
4	School Supplies	Calculator	user@gator.com	active	22:13:14.723175	none necessary
5	Electronics	Keyboard	user@gator.com	active	22:14:50.479309	none necessary
10	Electronics	USB Hub	user@gator.com	active	22:24:55.389012	none necessary
7	Books	Piano Book	user@gator.com	active	22:17:23.10269	none necessary
6	Books	Beginner Guitar Book	user@gator.com	active	22:16:06.2851	none necessary
9	Electronics	Computer Mouse	user@gator.com	active	22:22:28.102831	none necessary

## About Page

SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only  
Section 01 Team 03



All Categories ▾

Search EduGator

[Post](#)

[My Account](#)

[Admin](#)

**[About](#)**

[Login](#)

- [Chris Johansen](#)
- [Ivan Varela](#)
- [Kenneth Surban](#)
- [Liyao Jiang](#)
- [James Andrews](#)
- [Ian Dennis](#)
- [Shirin Namiranian](#)

# Database Screen Shots

## Items Table

```
team03-648::DATABASE-> select * from items;
-[ RECORD 1 ]-----
post_id      | 3
email        | user@gator.com
name         | Iphone Charger
category     | Electronics
description   | An Iphone charger with a lightning cable. Pretty much fresh out of the box.
price        | 12.55
condition    | Like New
image_path    | https://edugator-items.s3.us-west-1.amazonaws.com/1545430204349
created_at   | 22:10:05.101475
thumbnail    | https://edugator-items.s3.us-west-1.amazonaws.com/1545430204589
approval_status | active
-[ RECORD 2 ]-----
post_id      | 2
email        | user@gator.com
name         | IClicker 2
category     | School Supplies
description   | An IClicker you can use for your lectures. I only used this for one class, and I'll even include the batteries.
price        | 15.45
condition    | Lightly Used
image_path    | https://edugator-items.s3.us-west-1.amazonaws.com/1545430083278
created_at   | 22:08:04.158226
thumbnail    | https://edugator-items.s3.us-west-1.amazonaws.com/1545430083575
approval_status | active
-[ RECORD 3 ]-----
post_id      | 4
email        | user@gator.com
name         | Calculator
category     | School Supplies
description   | A TI-84 calculator that got me through all of my math classes. It's still in really good condition.
price        | 35.75
condition    | Lightly Used
image_path    | https://edugator-items.s3.us-west-1.amazonaws.com/1545430393890
created_at   | 22:13:14.723175
thumbnail    | https://edugator-items.s3.us-west-1.amazonaws.com/1545430394198
approval_status | active
-----
post_id      | 5
email        | user@gator.com
name         | Keyboard
category     | Electronics
description   | USB keyboard that functions really well. Just a little bit dusty.
price        | 15.99
condition    | Well Used
image_path    | https://edugator-items.s3.us-west-1.amazonaws.com/1545430489648
created_at   | 22:14:50.479900
thumbnail    | https://edugator-items.s3.us-west-1.amazonaws.com/1545430489958
approval_status | active
-----
post_id      | 10
email        | user@gator.com
name         | USB Hub
category     | Electronics
description   | Ever have the problem that your laptop doesn't have enough usb ports? Now you won't.
:
```

## Categories Table

```
team03-648::DATABASE=> select * from categories
team03-648::DATABASE-> ;
      name
-----
Electronics
Furniture
Books
School Supplies
Appliances
(5 rows)
```

## Users Table

```
team03-648::DATABASE=> select * from users;
-[ RECORD 1 ]-----
email      | admin@admin.com
password   | $2b$08$5e1qP8c0ZBJ3lFdoqFNh4ea4L7SCLsfH1irr9J0p/cX2k8NocyHEe
user_id    | 193
roles      | admin
-[ RECORD 2 ]-----
email      | test123@gmail.com
password   | $2b$08$pwMoMtV.90jQr.XHcsJnbeQqYaRvn7UMh6Izes6SmoL6MVlThMxGK
user_id    | 196
roles      | user
-[ RECORD 3 ]-----
email      | student@mail.sfsu.edu
password   | $2b$08$g1DPjF6kXB11HSriQY4ms0AUUnJ/f1M3xgLA452ac4.DoF7uRiGTN.
user_id    | 198
roles      | user
-[ RECORD 4 ]-----
email      | user@gator.com
password   | $2b$08$q57oDddy55LyBDyZE7UVU0u.lRDDwHMr1s0IYiEYW0EFNF1d9UDK6
user_id    | 195
roles      | user
-[ RECORD 5 ]-----
email      | testtest@mail.com
password   | $2b$08$CZ590Rh0XsG8ZKnmPexD.ucsI3ZeWVMtY5bCTsKp0X7.JkaCRfccW
user_id    | 197
roles      | user
```

## Google Analytics Plot

Introducing Google signals BETA

Unlock new cross-device capabilities and more. [Get Started](#)

All

Favorites

Recents

Search

Visit Platform Home

Analytics Accounts

Properties & Apps

Views

James Andrews  
131155145

Jamesftw  
55411498

<https://team03-648.herokuapp.com/>  
 UA-131155145-1

All Web Site Data  
186815265



# Team Member Contributions

## Team 03 CSC 648 Reflection email



James Howard Andrews

Sat 12/22/2018 11:53 AM

To: Ian Dennis; johansen@sfsu.edu; Ivan Antonio Varela; Kenneth Clark Asencion Surban; liyaojiang@my.smccd.edu; Shirin Namiranian



Reply all | v

James Andrews - Team Lead

Project Contributions:

- As the team lead, I attended team meetings for both the frontend and backend teams, as well as scheduling all team meetings every Wednesday.
- Outlined the general architecture for the project. (Client/Server).
- Relayed feedback and messages from Professor Petkovic to the team.
- Added the backend boilerplate functionality for posting images to S3.
- Help other team members with technical questions.

Github Contribution numbers:

86 commits.

## Team 03 CSC 648 Reflection email



Ivan Antonio Varela

Sat 12/22/2018 12:04 PM

To: James Howard Andrews; Ian Dennis; johansen@sfsu.edu; Kenneth Clark Asencion Surban; liyaojiang@my.smccd.edu; Shirin Namiranian



Reply all | v

- Front-end lead
- Learned react from scratch
- implemented about page
- implemented navigation bar
- implemented search page
- implemented sorting by price
- implemented dropdown categories in the nav bar
- implemented product cards
- Implemented admin dashboard
- implemented search functionality

GitHub contributions to master branch: 41 commits

\*\*\*

## Team 03 CSC 648 Reflection email



Ian Dennis

Sat 12/22/2018 12:23 PM

To: Ivan Antonio Varela; James Howard Andrews; johansen@sfsu.edu; Kenneth Clark Asencion Surban; liyaojiang@my.smccd.edu; Shirin Namirani



Reply all | v

Role: Backend Lead

Contributions:

- Implemented back-end portion of the vertical prototype
- Added back-end thumbnailing
- Added the categories table and associated routes
- Populated the items table
- Organized weekly back-end team meetings and tasks
- PR reviewer for back-end, enforced coding standards
- Formatted Milestone documents, and worked on various sections of the different milestones

Github Commit Count: 16

\* Note: We started to "Squash and Merge" features into master after the vertical prototype, which skews team member commit numbers after M1.

...

## Team 03 CSC 648 Reflection email



Christopher John Johansen <johansen@sfsu.edu>

Today, 12:31 PM



Chris Johansen: Backend Dev

Project Contributions:

- Implemented user sessions using passport.js
- Implemented JWTs for protected routes
- Set up database tables
- Managed the database
- Created admin approval routes
- Updated messaging routes
- Attended all team and sub-team meetings
- Helped fix issues through communicating on slack

GitHub contributions to master branch: 6 - after squashing and merging commits

...

## Team 03 CSC 648 Reflection email



Kenneth Clark Asencion Surban

Sat 12/22/2018 12:34 PM

To: James Howard Andrews; Ian Dennis; johansen@sfsu.edu; Ivan Antonio Varela; liyaojiang@my.smccd.edu; Shirin Namiranian



Reply all | v

**Kenneth Surban - Frontend**

**Project Contributions:**

- learned React
- implemented larger view for item
- implemented User dashboard
- implemented Post form
- implemented posting functionality
- implemented message form
- implemented messaging functionality

**Git Contribution numbers:**

- 8 commits

...



Shirin Namiranian

Sat 12/22/2018 2:18 PM

To: Kenneth Clark Asencion Surban; James Howard Andrews; Ian Dennis; johansen@sfsu.edu; Ivan Antonio Varela; liyaojiang@my.smccd.edu



Reply all | v

**Shirin Namiranian: FrontEnd**

**Project Contributions:**

- Learned React.js by myself
- Implemented Login page and checking all error handling
- Implemented Signin page and checking all error handling
- Implemented Forgot password page
- Implemented Term and Conditions page
- Implemented PostMessage page But Kenneth helped with choosing the right Api and error handling
- Implemented messaging for UserDashboard page That Kenneth already was started it and I added rows and some functions.
- Created logout function
- Reordered Navbar to login at the last Nav item and shows logout
- Implementing Routes:
  1. Created history that allows us to redirect users to other pages
  2. Created a custom PrivateRoute component that checks whether or not the user has an auth token in their local storage, if so they have access to admin, post, my account pages, if not, they get redirected to the login page if they click on any of those pages.

**Git contribution numbers:** 1Commit



Liyao Jiang <liyaojiang@my.smccd.edu>

Today, 1:40 PM



Liyao Jiang- Backend

Project Contributions:

- Implemented Post items to database
- Implemented Post tutor services and coursebooks to database.
- Implemented get and post routes for message

Git contributions numbers:

2 commits, 167 additions

Best,

Liyao Jiang

...

## Post analysis

Building Edugator has been a fun and rewarding task over this past semester. As the team lead, I can say that I am proud of what we were able to accomplish in the amount of time given. Especially being that we are also students with other classes and obligations added on top of the class. Other than a few hiccups with the communication between user login on the frontend and the backend, our team ran relatively smoothly. I feel this is because of our clear distinction of responsibilities in our applications architecture and constant communication on Slack. This combined with one weekly all-team meetings, one weekly meeting for the frontend, and weekly meeting for the backend team help us all stay on task and identify obstacles early so we could fix them before they become a bigger issue.

As a team, we used Github projects for our project management, but further into the project the task board went unused. If I were to change one thing about our team structure and project management, I would make a more detailed version of our task board. I would break bigger task down into smaller more digestible task that can be done quickly. Other than that I feel our team did well.

In conclusion, Team 03 learned a lot about software development from the initial case study and persona phase all the way up to implementation and migration of code on to our Heroku server.