

SW Engineering CSC 648/848 Summer 2016**“ShopU”****Group 3***Local Group*

Matt Kindblad

SFSU Student[\(mkindblad@hotmail.com\)](mailto:mkindblad@hotmail.com)

Tyler Willis

SFSU Student

Qihong Kuang

SFSU Graduate Student

Luis Estrada

SFSU Student

Panawat Sanitchon

SFSU Student

Valerio Zhang

*McGill University/SFSU Student***Milestone 2**

7/15/16

Version	Date	Description
1	7/11/16	First draft
2	7/13/16	1st revision from
comments		
3	7/15/16	2nd revision from
scrum		

Table of Contents

- [1. Executive Summary](#)
- [2. Use Cases](#)
 - [2.1 User Guests](#)
 - [2.2 User Seller](#)
 - [2.3 User Buyer](#)
 - [2.4 Administration](#)
- [3. Data Definitions](#)
- [4. Functional specifications](#)
 - [4.1 Priority 1](#)
 - [4.2 Priority 2](#)
 - [4.3 Priority 3](#)
- [5. Non-functional specifications](#)
- [6. High-level UI Mockups](#)
 - [6.1 Home page](#)
 - [6.2 Registration page](#)
 - [6.3 Sign in page](#)
 - [6.4 Search results page](#)
 - [6.5 Item details page](#)
 - [6.6 Upload items page](#)
- [7. Storyboards](#)
 - [7.1 Buyers \(unregistered and registered\)](#)
 - [7.2 Sellers](#)
 - [7.3 Admin](#)
- [8. Competitive analysis](#)
- [9. High-level system architecture](#)
- [10. High-level UML Diagrams](#)
 - [10.1 Class UML](#)
 - [10.2 Component & Deployment UML](#)
- [11. High-level APIs](#)
 - [11.1 Registered User](#)
 - [11.2 Order](#)
 - [11.3 Item](#)
 - [11.4 Admin](#)
 - [11.5 Database](#)
- [12. Key Risks](#)
- [13. Team Organization](#)
- [14. Attachments](#)

1. Executive Summary

ShopU is an electronic commerce website that provides consumer-to-consumer sales services exclusively to students attending San Francisco State University (SFSU). Our website focuses on merchandise sales for students, and provides the option to sell services such as tutoring from student to student. This website is built on a concept that provides a quick way for students to list products for sale based on categories, and provides for free advertisement services. Students will be able to browse goods and services listed based on category, and will be able to filter by price range as well as by other features that fit their budget and preferences, which provides convenience to students who have busy schedules to search for and add products to earn that extra money they need.

Our startup team consists of developers with relevant experience in website development. Our team leader Matt has experience with database schemas, SQL and PHP; supported by our CTO Tyler who specializes in managing databases; and Luis who also has experience with databases and PHP. Our front-end developers Panawat, Qihong, and Valerio all have experience with UI development and languages CSS, Bootstrap, JavaScript & jQuery.

Our design focuses on user experience, making users' browsing experiences as familiar and as simple as possible. A student will be able to register with ease just by providing information such as their student ID number, as well as their shipping address, which they can save for future purchases. As for payment methods, students will be able to pay for transactions through trusted payment services such as PayPal, or by other means, i.e., with a bank account, check or credit card. In addition, our website provides on-site messaging for students to make arrangements for a meeting on-/off-campus, and a way for buyers to contact sellers on any questions regarding the products or services. Lastly, students who purchase products will each receive an email confirmation containing details of the products or orders that have been placed.

Our website is supervised by administrators who are responsible for overseeing user profiles, providing customer support services through on-site messaging, and filtering inappropriate products or services on our website. This responsibility aims to fulfill student expectations and provide for a great customer experience and overall satisfaction.

2. Use Cases

2.1 User Guests

Casey is a current SFSU student from Oakland. Being a college student has been hard on his financial life, so he goes online to find a website where he could potentially find cheap items and services that he may want or need, and that will help him get through the college years. Being an SFSU student, he has heard of **ShopU**, so he visits the website and immediately comes to a screen that shows the **ShopU logo**, a **search bar** with **categories**, a **filter** list, and promoted **items**, including used school books and furniture within reasonable prices. He scrolls through the categories and finds the **services** section, and is interested to see that tutoring is a service offered by **registered users** on the site. Casey thinks of something in particular he wants, and then searches for that item/service. He is presented with multiple pictures of the item, and can also read detailed information added by the **seller**.

Deciding that he's interested in a certain item, he selects it and comes upon a **detailed view** with information on the item such as price, condition, and location. He likes the product and decides to order it to add to his **shopping cart**, but is told that he is an **unregistered user**, and must **sign up** in order to buy and sell items on the website. After filling out some basic information (his name, student ID and email address) and agreeing with the site's privacy policy as well as terms & conditions, Casey creates his own ShopU account and becomes a registered user on our website. He can upload a **profile picture**, and can edit or delete any information on his profile at anytime. He bundles his bank account or PayPal with his ShopU account to be able to withdraw earnings on our website and make transactions with other users.

2.2 User Seller

Andy is a student at SFSU. Recently, he's been busy preparing for his graduation, which includes his senior presentation, class projects and job search. He also wants to dispose of some of his furniture, books, and his bicycle in his dorm. Through the strong recommendation of his roommate, he decides to sign up on our company's website so that he can sell his items to other SFSU students and make extra money.

He first categorizes the items he wants to post online and takes some nice, clear pictures for each one. He chooses the appropriate category on the uploading page and then uploads the item pictures. He may upload one or more pictures for each item and can decide on an appropriate price for his items. Andy also adds some key descriptions (metadata) so that **buyers** can get more information about each item, such as their location, size, and weight. Other registered buyers are interested in Andy's items and place their **orders** for them. A message is sent by the system automatically to let Andy know that someone has placed an order on his items, containing the name, shipping address, and contact information of the buyer, as well as the name and quantity of each item purchased. Andy prepares his shipment once he gets the message, and enters the tracking number to the system so the buyer can track the shipment status.

2.3 User Buyer

Sarah is an SFSU freshman who comes from Dallas, Texas and lives near campus. She wants to purchase a cheap bike to travel between her apartment and school since she has a limited budget. She decides to sign up with ShopU. Using the search bar, categories, and filter, she browses through an assortment of potential bikes. After adding a bike that meets her criteria to her shopping cart, Sarah clicks the bright and well-marked button to **checkout** and purchase the item. On the payment page, Sarah needs to fill out her name, phone number and shipping address. She can save her address if she wants so she won't have to retype it for future purchases on our website. Along with this information she provides, she also needs to provide her payment information, such as a credit card, bank or PayPal account.

After placing an order, Sarah receives an automatic confirmation message from our website. The email contains the item names, the price of each item, and the total price of the entire order. After the seller ships their item, Sarah receives another email to let her know that her order has been shipped. The email contains the shipped item, the price, and the tracking number. She can check the order status by accessing her ShopU account.

2.4 Administration

Amy is an SFSU alumna from San Francisco who works as an **admin** for ShopU. With this responsibility, she has executive control over the website and its edit privileges, as well as privileges with the server not accessible to a regular user. Amy can **sign in** to her account using her administration credentials, and can delete user accounts upon request. She has the power to warn the user, or even ban them from the website and remove their profile. She can also do the same with items and services posted on ShopU that do not meet the terms & conditions the users agreed to.

In addition to this, Amy has the ability to check the market analytics of the site, and share this knowledge with the other head members of ShopU. And since ShopU is a startup and a small company, she will help with customer support, essentially acting at a level higher than regular customer support.

3. Data Definitions

ShopU - The name of our company website.

ShopU Logo - This will display a .jpeg image of the team project logo.

Admin – Short for Administrator. They have access to databases, and can remove content/users as necessary based on terms & conditions. Admin information is created during registration of an admin and stored in the database.

Unregistered User - This user can browse the website's items and services but has not registered. They will not be able to purchase or request items or services until they register.

Registered User - This user will be able to sign into their account on the website, having registered their account with their name, email address, and student ID, and having agreed to the privacy and terms & conditions agreements. This user will be able to buy and/or offer items/services to other users, and receive communications from ShopU about transactions, as well as message other users (priority 2). Registered user information is created during registration process and stored in the database.

User – Generic term that represents both Unregistered Users and Registered Users.

Buyer – A registered user who is buying an item/service in an order process.

Seller – A registered user who is selling an item/service in an order process.

List Field - The area on the homepage where items are listed.

Detailed View - When the user clicks the item or service, they will be taken to a page dedicated to more specific information about the item or service such as price, condition, location, as well as additional picture options. They will also have the option to purchase the item/service from here.

Categories - Categories to simplify the user's search, such books, furniture, electronics, clothes, other, etc. It will restrict search results to items and services that fall within that category.

Search Bar - The user can search for a specific item or service, as well as metadata tags by entering information into the search bar and clicking enter.

Magnifier Icon - The icon the right of the search bar that can be pressed to submit a search.

Sign Up Button - This link will be displayed for unregistered users who are using our website so they may sign up to ShopU.

Sign In Button – A link displayed to any user not signed in. It will allow unregistered users to sign up using our registration process and become a registered user.

Sign Out Button - This button will be displayed for registered users who are currently signed in so they may sign out.

Buy Button - This button will be next to every item offered to add to the shopping cart.

List It Button - This button will be used to upload an item when in the upload view.

Checkout - This button will take the registered user to the checkout process where they may complete their transactions.

Shopping Cart - This will be a button link that takes the user to a list of currently ordered products that have not been checked out yet (priority 2).

Filter - Functionality to refine searches by filtering out metadata, features, etc.

Item - General description of an item or product to be sold/purchased. Item created when a seller uploads it to our website, and is stored in the database as an item listing.

Price - The cost of an item; can be used for filtering.

Condition - The condition of an item. A searchable keyword as well as can be used for filtering.

Author - The author of a book; a searchable keyword.

Publisher - The publisher of a book; a searchable keyword.

Publish Date - The date book was published; can be used for filtering.

Release Date - The date that an item was originally released for sale.

Weight - The weight of an furniture item; can be used for filtering.

Size - The size of an item (for clothing, S, M, L, XL, etc.), and dimensions for furniture. A searchable keyword as well as can be used for filtering.

Service - General description of a service needed or provided. Service created when a seller uploads the service to our website, and it is stored in the database as a service listing.

Order - Represents a transaction for an item or service. Selected item/service the user wishes to purchase are added to the shopping cart with other items ordered. Order is linked to a buyer and seller whom are registered users, and an item/service. Order created when a transaction occurs by a buyer clicking the buy button. Until shopping cart is implemented, orders will not be stored.

Checkout - All user purchased items requested to order.

Profile Picture - A registered user will be able to upload a profile picture that will be posted in the corner, modified to a thumbnail size. This thumbnail will allow the user to see their profile content and settings. Profile picture created when a registered user uploads it, and is stored along with the registered user information on the database.

4. Functional specifications

4.1 Priority 1

4.1.1 Unregistered Users

1. Users shall get the added capabilities of Registered Users once they register an account through the application.
2. Users shall only be able to register if they are SFSU students.
3. Users shall register with a username, email address, student Id, and password.
4. Users shall agree with the website's privacy policy and its terms & conditions when registering.
5. Users shall be able to search for items.
6. Users shall be able to refine search results by different categories.
7. Users shall be able to browse through items from the main list field.
8. Users shall be able to view more information about items by selecting them, including specific details, condition of item, and additional pictures.

4.1.2 Registered Users

9. Users shall be able to buy/sell items.
10. Users shall be able to upload information about their items, including descriptions, categories, keywords, and photos.
11. Users selling an item shall receive a message when that item is purchased, containing the name, shipping address, and/or contact information of the buyer as provided by the buyer, as well as the name of the item purchased.
12. Users buying an item shall receive a message when they place an order, containing the item name, the price of the item, and the total price of the order.

4.1.3 Administrators

13. Administrators shall be able to monitor content on the website by receiving information of newly registered users and uploaded items/services.
14. Administrators shall be able to delete content if inappropriate and/or in violation of the terms & conditions..
15. Administrators shall be able to ban/remove users if they violate the terms & conditions.

4.1.4 Application

- 16. Application shall be able to categorize listings.
- 17. Application shall display promoted items on the homepage.

4.2 Priority 2

4.2.1 Registered Users

- 18. Users shall be able to buy/offer services.
- 19. Users shall be able to update their profiles, including uploading a profile picture and adding some personal information.

4.2.2 Administrators

- 20. Administrators shall be able to provide customer support to users.

4.2.3 Application

- 21. Application shall have a method of communication via messaging to contact parties.
- 22. Application shall have a shopping cart to store multiple items ordered.
- 23. Application shall allow registered users to checkout items from the shopping cart.

4.3 Priority 3

4.3.1 Registered Users

- 24. Users shall be able to bundle their bank or PayPal account with their ShopU account to facilitate making transactions if desired.
- 25. Users buying an item shall receive a message when their order has been shipped, containing the item names, the price of each item, the total price of the order, and the tracking number of the order.

4.3.2 Application

- 26. Application shall provide market analytics.

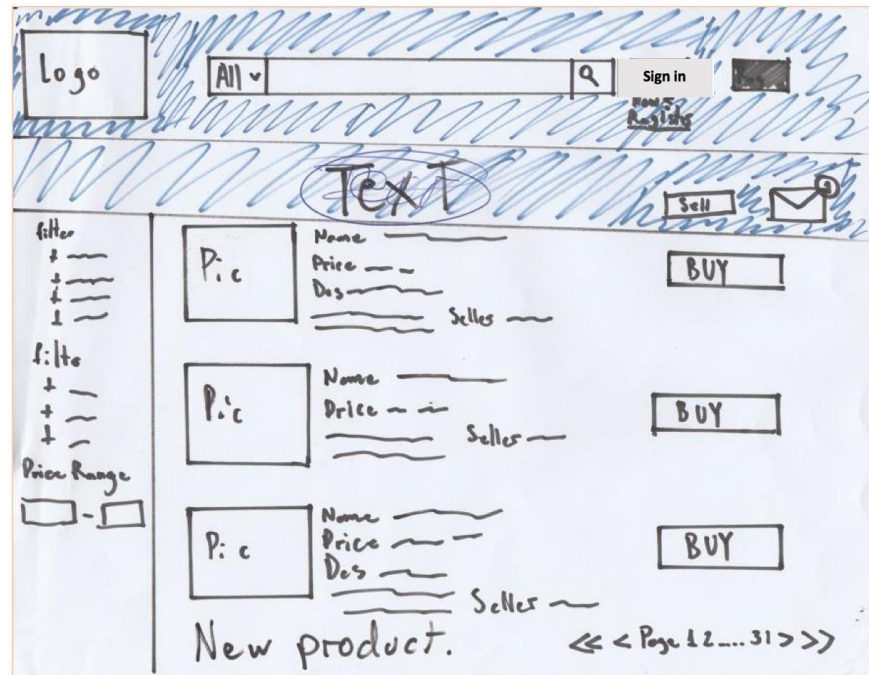
5. Non-functional specifications

1. Application shall be developed using class-provided LAMP stack.
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks shall be explicitly approved by Marc Sosnick on a case-by-case basis.
3. Application shall be hosted and deployed in the cloud via Amazon Web Services as specified in the class.
4. Application shall be optimized for standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Firefox, Safari, Chrome, Opera, and IE.
5. Application shall degrade nicely for different-sized windows using class-approved programming technologies and frameworks.
6. Data shall be stored in the database on the class server in the team's account.
7. Application shall be served from the team's account.
8. No more than 50 concurrent users shall be accessing the application at any given time.
9. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users via the user registration process.
10. The language used shall be English.
11. Application shall be very easy to use and intuitive through careful UI design. No prior training shall be required to use the website.
12. Google analytics shall be added for major site functions.
13. Pay functionality (how to pay for goods) shall be simulated with proper UI, no backend.
14. Messaging between buyers and sellers shall only occur with SFSU e-mail addresses on the sfsuswe server for both sending and receiving messages.
15. No private information will be released without user consent for transactions.
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development, using the tools approved by instructors.
17. The website shall prominently display the following text on all pages "SFSU/FAU/Fulda Software Engineering Project, Summer 2016. For Demonstration Only." (Important so as to not confuse this with a real application).
18. Application shall have security measures to ensure only SFSU students are using it (e.g. SFSU email, or verifying student Id).
19. Browser search results will have a response time of 1-2 seconds.

6. High-level UI Mockups

Listed below are the rough draft views of the UI for the ShopU website. The copyright disclaimer on all pages includes the text “SFSU/FAU/Fulda Software Engineering Project, Summer 2016. For Demonstration Only.”

6.1 Home page



This is the view that greets users to the homepage. The search bar is on the top center of the page and the category is next to the search bar on the left. Users can see the ShopU logo on the top left of our website. The buttons to sign in and sign up are to the right of the search bar. Below the search bar, we'll put some text to inform the users of the business purpose of ShopU.

6.2 Registration page

ShopU

Create account

Username

SFSU ID

Email

Password

Password again

☐ Create your ShopU account

By creating an account, you agree to ShopU's [Conditions of Use](#) and [Privacy Notice](#)

Already have an account? [Sign in](#) ▾

[Conditions of Use](#) [Privacy Notice](#)

© copyright

This screen is brought up when a user is prompted to sign up, which either occurs when the user requests to sign up or when an unregistered user attempts to place an order.

6.3 Sign in page

ShopU

Signin

SFSU ID\Username

Password

//// Sign in ////

— New to ShopU —

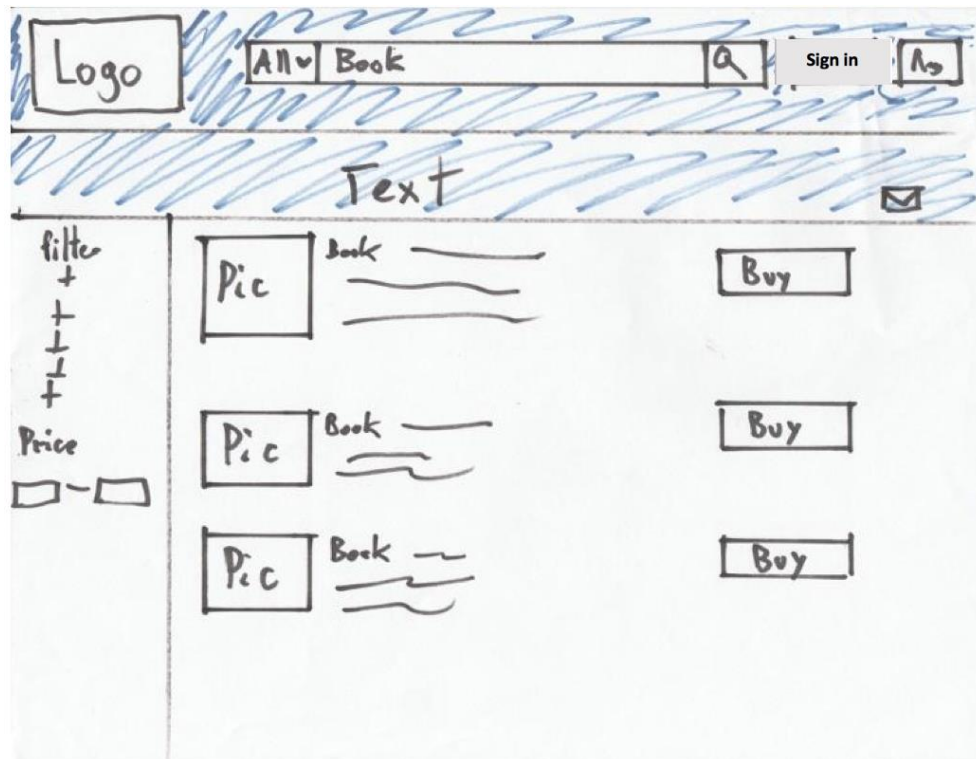
/// Create your ShopU account ///

Condition of Use Privacy notice

© copyright

Basic sign in page

6.4 Search results page



Above is the 'search results' page. Say if a user types "Book" into the search bar, the matched results will be listed from top to bottom in the list field. Users can see a thumbnail of the item and simple details about the item. A "Buy" button is next to the item description.

6.5 Item details page

logo All sign in

Text

pic Name: --
Year: --
Price: --
Detail: --

Qty
Price: --

© copyright.

If users click on a picture of an item on the 'search results' page, this 'item details' page will appear.

6.6 Upload items page

Below is a view for when an SFSU student wants to upload an item for sale.

The wireframe illustrates a 'seller upload page' with the following components:

- Header:** Includes a 'Logo' box, an 'Allv' label, a search bar with a magnifying glass icon, and a 'Signin' button.
- Title:** The page is titled 'Text (seller upload page)' in a handwritten style.
- Form Fields:**
 - Three 'click to upload pic' buttons are positioned at the top of the main content area.
 - 'category : Text' with a text input field.
 - 'Item name: Text' with a text input field.
 - 'Your price: Text' with a text input field.
 - 'condition: Text' with a text input field.
 - 'Details : Text' with a larger text input area.
- Terms and Conditions:** A checkbox is followed by the text: 'By clicking "List it", you agree to ShopU's Condition of Use and Privacy Notice'.
- Action Button:** A 'List it' button is located at the bottom right of the form area.
- Footer:** Contains the text 'Condition of Use', 'Privacy Notice', and '© copyright.'

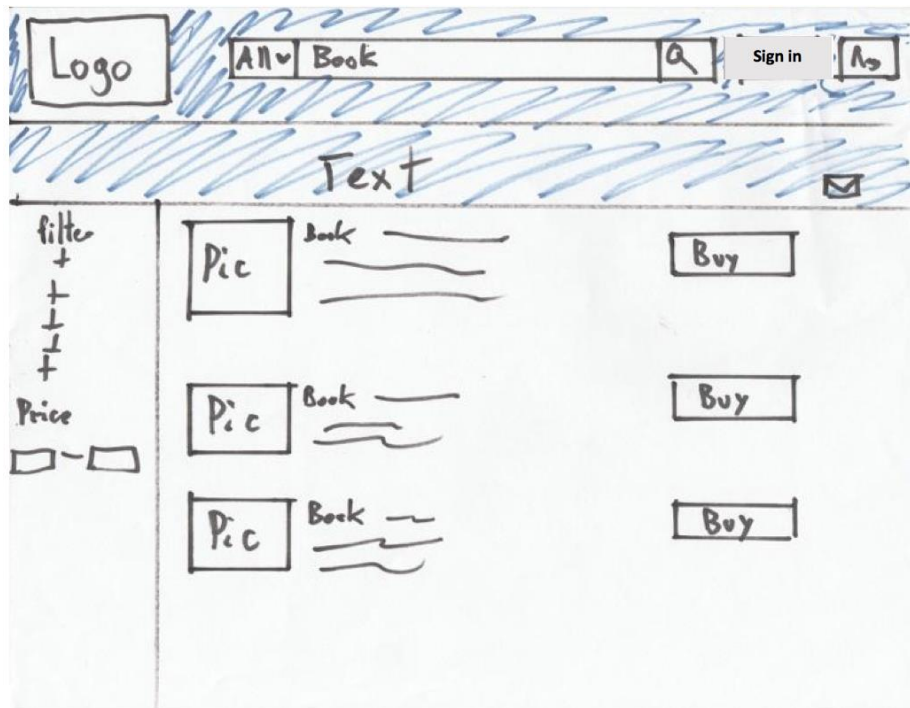
7. Storyboards

7.1 Buyers (unregistered and registered)

Below is a storyboard of the views an SFSU student would go through as they visit the site, see items they like, attempt to place an order, and then create an account on the website.

7.1.1 Visit website without signing in

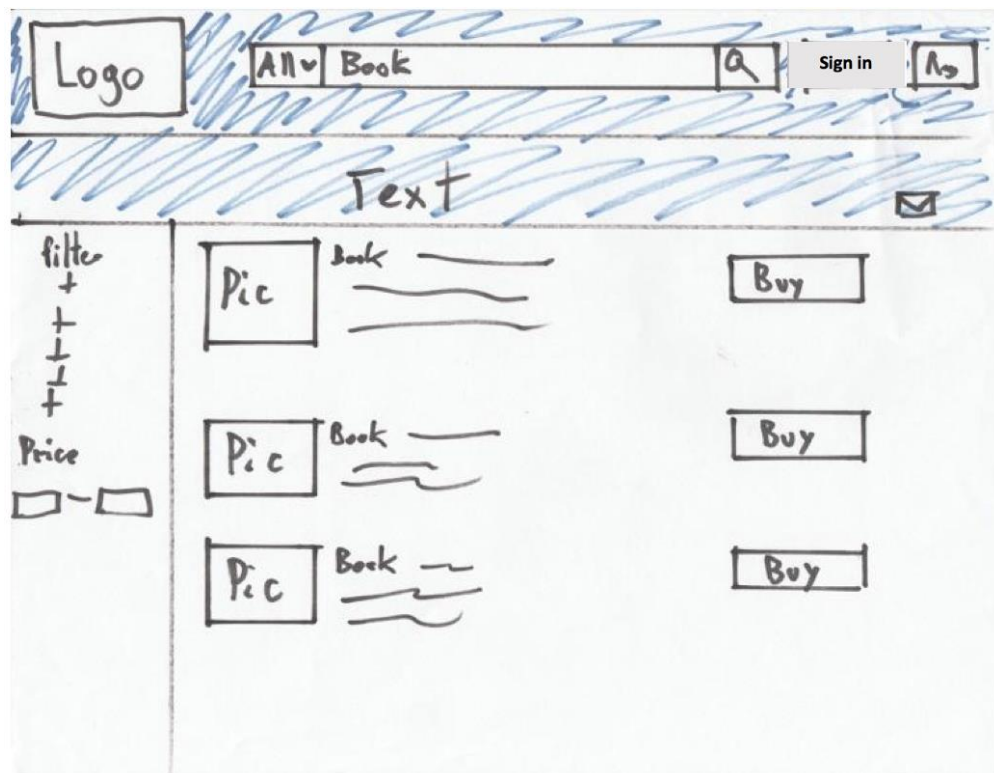
First, this student visits our home page by typing the website's URL into their browser.



This unregistered user can see our website logo on the top left of the homepage. In the top middle part of the homepage are the search bar and categories. The 'sign in' and 'sign up' buttons are next to the search bar. Below the search bar will be a section of text in order to inform the student of our website's business purpose. On the homepage, the latest uploaded items are listed in the list field and the filter options are on the left side of the list field. The student can search for the items they want by typing into the search bar. To narrow the scope of searches, the student buyer can also click the small arrow beside the category "All" and select a more specific category. They can press "Return" on their keyboards or click the magnifier icon to the right of the search bar to get search results.

7.1.2 Search results listing

Say if the student buyer types “Book” in the search bar and presses “Return” on their keyboard, items that match the keyword “Book” are listed. The search results page looks very simple so that it’s easy for the student to read the content. The “Buy” button is next to the description of the item and the filter function is on the left side of the page. The student can filter out the items based on price and other criteria.



The student buyer clicks a picture of the first item. Our website redirects them to the 'detailed view' page. On this page, users can look at several item pictures, prices, and details, and they can also place an order.

logo All sign in

Text

pic

Name:

Year:

Price:

Detail:

Qty

Price:

© copyright.

Detailed view

7.1.3 Sign in

The student buyer likes the first item on the search page so they clicks “Buy.” Since they haven’t signed into their account, our website will redirect them to the sign in page. If this student buyer is an existing user of our website, they just need to type in their SFSU ID and password to access their account and place their order.

ShopU

Signin

SFSU ID \ Username

Password

//// Sign in \\\

— New to ShopU —

/// Create your ShopU account ///

Condition of Use Privacy notice

© copyright

7.1.4 Sign up

If this student buyer hasn't created an account on our website, they can and go to the 'signup' page to register. They need to fill in their username, SFSU ID, email and password, as well as agree to our Privacy Policy and Terms & Conditions before clicking on "Create your ShopU account," finishing the registration process.

ShopU

Create account

Username

SFSU ID

Email

Password

Password again

By creating an account, you agree to ShopU's Conditions of Use and Privacy Notice

Already have an account? [Sign in](#) ▼

[Conditions of Use](#)[Privacy Notice](#)

© copyright

7.2 Sellers

Below is a storyboard of the views a student seller would go through as they upload an item on our website.

7.2.1 Sign in

First they need to sign in to their account. If they are new to our website, they first need to create an account.

The storyboard shows a hand-drawn sign-in page for 'ShopU'. The page is titled 'ShopU' at the top. Below the title is a rectangular box containing the following elements:

- The word 'Signin' in the top left corner.
- A label 'SFSU ID \ Username' above a text input field.
- A label 'Password' above another text input field.
- A button labeled 'Sign in' with diagonal hatching on either side.
- A link labeled 'New to ShopU' with horizontal lines on either side.
- A button labeled 'Create your ShopU account' with diagonal hatching on either side.

Below the sign-in box, there is a horizontal line, followed by the text 'Condition of Use Privacy notice' and '© copyright'.

Sign in page

ShopU

Create account

Username

SFSU ID

Email

Password

Password again

//Create your ShopU account//

By creating an account, you agree to
shopU's Conditions of Use and Privacy Notice

Already have an account? Sign in ▼

[Conditions of Use](#)

[Privacy Notice](#)

© copyright

Registration page

7.2.2 Upload an item

After they sign in to their account, they may upload an item to our website. Below is the view they see when they try to upload an item.

The sketch shows a web page layout for uploading an item. At the top, there is a navigation bar with a 'Logo' box, an 'Allv' box, a search bar with a magnifying glass icon, and a 'Signin' button. Below this is a title 'Text (seller upload page)' with decorative wavy lines. The main content area is enclosed in a large rectangle and contains three 'click to upload pic' buttons at the top. Below these are four text input fields labeled 'category:', 'Item name:', 'Your price:', and 'condition:'. The 'Details:' label is followed by a larger text area. At the bottom of the form, there is a checkbox with the text 'By clicking "List it", you agree to ShopU's Condition of Use and Privacy Notice'. To the right of the checkbox is a 'List it' button. Below the main form area, there are links for 'Condition of Use' and 'Privacy Notice', and a copyright notice '© copyright.'

On the upload page, the seller needs to provide the name, category, price, description, and pictures of the item. Additionally, they need to check the box indicating that they agree to our Privacy Policy and Terms & Conditions before clicking “List It” to finish uploading their item.

7.3 Admin

Below is a storyboard of the views an admin would go through as they manage the information of users.

7.3.1 Sign in

First, the admin needs to sign into their administration account through the sign in page.

The storyboard shows a hand-drawn sketch of a sign-in page for 'ShopU'. The page is titled 'ShopU' at the top. Below the title is a rectangular box containing the following elements:

- The word 'Signin' is written at the top left of the box.
- Below 'Signin' is the text 'SFSU ID\Username'.
- Underneath the text is a horizontal input field.
- Below the input field is the text 'Password'.
- Underneath the text is another horizontal input field.
- Below the password field is a button labeled 'Sign in' with diagonal hatching on either side.
- Below the button is the text 'New to ShopU' with horizontal lines on either side.
- At the bottom of the box is a button labeled 'Create your ShopU account' with diagonal hatching on either side.

Below the sign-in box, there is a horizontal line, followed by the text 'Condition of Use Privacy notice' and '© copyright'.

7.3.2 Admin management

After they sign in, the website redirects them to the admin page.

User name	SPSU ID	Email	Item order	Status
~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~	~~~~~	~~~~~

On the admin page, administrators can search for user or item information. Administrators can also send and receive messages to and from other users.

## 8. Competitive analysis

Feature	Amazon.com	Ebay.com	HBX.com	Craigslist.org	ShopU.com
Human Services	-	-	-	+	+
Customer Service (Live Chat)	++	-	-	-	++
Search	++	++	++	-	++
Better Design, Less Clutter, Filter picture resolution	++	++	++	+	++
Shopping Cart	++	++	+	-	++
Analytics (Market Price)	+(for Sellers)	-	-	-	++
Homepage Trending & Recommended Items	++	+	+	-	+

- : Means missing the feature, or below market standards.
- + : Means has the feature, or, on-par with market standards.
- ++ : Means has the feature and above-par with market standards.

The initial market analysis shows that amazon.com, ebay.com, hypebeast's hbx.com, and craigslist.org are the biggest competitors in the e-commerce/e-classifieds markets. However, niche marketing and implementation towards only SFSU affiliates will allow this project to be on the vanguard of local e-commerce/e-classifieds markets. Mainly our focus is on user experience. ShopU will offer services as well as items that students can exchange (such as tutoring) using better UI and better browsing capabilities than Craigslist, thereby expanding our market reach and making our niche marketing strategy even more impressive. Additionally, to create a cohesive e-commerce environment which users will return to use, we must have a good customer support service. This will be done by creating a live chat service where users can use to address their problems and get a return solution punctually. Another feature we hope to implement is to provide better market analytics to help users make better and more informed decisions on their purchases, thereby, giving a better user experience (less scams, more informed purchases, more savings, etc.). In addition, we will have trending and recommended items displayed on the homepage to encourage clicks and consumer traffic.

## 9. High-level system architecture

### Software Components

- Operating system: Linux
- HTTP server: Apache Web Server
- Database management system: MySQL
- Server-side programming language: PHP 5.5

Collectively known as the LAMP stack, the combination of Linux, Apache, MySQL, and PHP offers a stable and scalable platform upon which to build database-driven web applications.

**Deployment:** Amazon Web Services (AWS), a suite of cloud computing services offered by Amazon.com.

### Database Schema:

- item(id, title, seller_id (fk), price, category, description, keywords, image)
- user(id, sfsu_id, name, email, password, image)
- admin(id, name, email, password, image)
- order(id, buyer_id (fk), seller_id (fk), item_id (fk), price, date, details)

**Images:** JPG image format. 2 MB image size limit to maintain efficiency. User thumbnail image size is 100x100 JPG. Image thumbnail image size 50x50 in JPG. We are using Medium BLOBs to store our images in the database.

**Search architecture:** following the MVC pattern, form data will be sent to a controller from a view, which will then be sent to a model to query the database (using the LIKE operator), returning data from the database back to a view for a user to see and interact with. Searches can be made for any keywords pertaining to items stored in the database, the categories of which are books, electronics, furniture, clothes, transportation, and other.

### Frameworks:

- Bootstrap - an open-source framework for creating responsive mobile-first websites. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components. Using Bootstrap as a frontend framework ensures compatibility with the following web browsers: Chrome, Safari, Firefox, Opera, and Internet Explorer.
- MINI PHP - a very simple, open-source PHP framework that provides an MVC structure in which we'll develop our website.
- jQuery - technically a JavaScript library, jQuery will simplify any client-side scripting used for our website.

## jQuery plugins:

- DataTables - an open-source, customizable jQuery plugin providing pagination, instant search, and multi-column ordering to tables.¹

**IDE:** NetBeans - an open-source, multi-language Integrated Development Environment (IDE) that provides tools for editing, debugging, and deploying code.

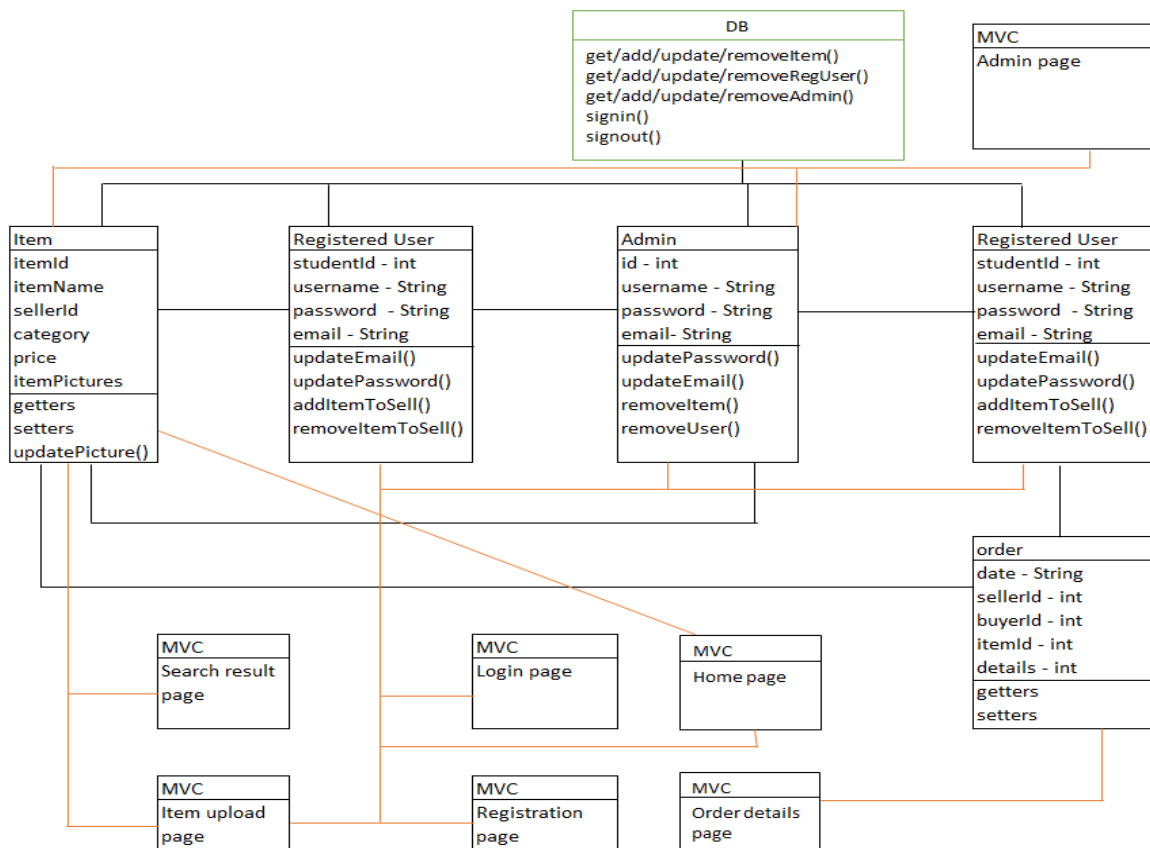
## Browser Version Support:

- Internet Explorer (IE11, IE10)
- Google Chrome (51.0.2704, 50.0.2661)
- Mozilla Firefox (47.0, 45.2.0esr)
- Safari (9.1, 8.0.8)
- Opera (37, 36)

**Version Control:** GitLab

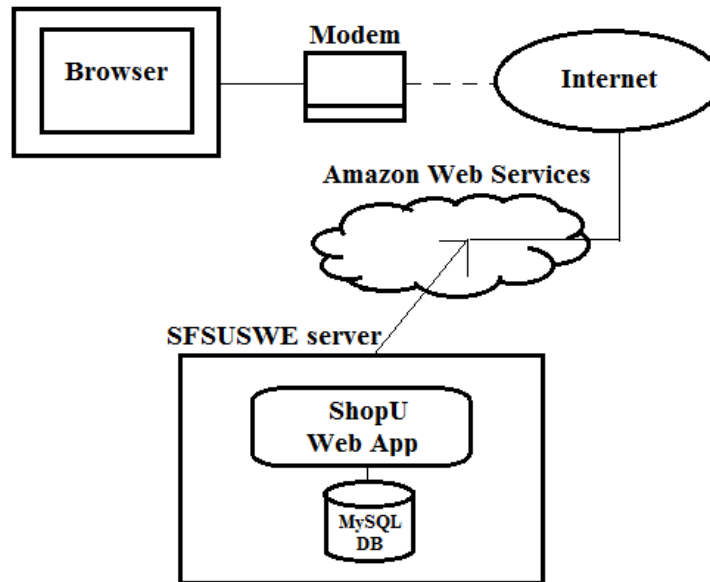
# 10. High-level UML Diagrams

## 10.1 Class UML



¹ See attached for permission to use this plugin.

## 10.2 Component & Deployment UML



## 11. High-level APIs

### 11.1 Registered User

Represents a single registered user in the directory of registered users.

RegisteredUser	
.username: String	Registered user's username.
.emailAddress: String	Registered user's email address.
.studentId: int	An unsigned integer value used to identify a registered user's SFSU student ID.
.profilePic: Image	An image bitmap that displays a user's profile picture.
.password: String	A secure string that enables a user to access his personal account.
.items: int[] (IDs)	Item(s) the registered user is currently selling.

.updatePassword(String): void	Enables a registered user to update their current password. <i>I/P: new password</i> <i>O/P: N/A</i>
.updateEmail(String): void	Enables a registered user to update their current email address. <i>I/P: new email address</i> <i>O/P: N/A</i>
.addItemIdToSell(itemId: int): void	Add an item to registered user's array of items to sell. <i>I/P: Id of new item user is selling</i> <i>O/P: N/A</i>
.removeItemIdToSell(int): void	Remove an item registered user is selling. <i>I/P: Id of item being sold to remove from user</i> <i>O/P: N/A</i>

## 11.2 Order

Represents a single transaction created when a buyer purchases an item.

Order	
.buyerId: int	Id of the buyer who places an order
.sellerId: int	Id of the seller who sells out the item
.itemId: int	Id of the ordered item
.totalPrice: float	Total price on the order
.orderDate: String/timestamp	Order date
.details: String	Order description and details
.getBuyerId(): int	Function used to get the buyer ID I/P: N/A O/P: <i>Id of the order's buyer</i>
.setBuyerId(int): void	Function used to set the buyer ID I/P: <i>Id of the registered user who is buying</i> O/P: N/A
.getSellerId(): int	Function used to get the seller ID I/P: N/A O/P: <i>Id of the order's seller</i>
.setSellerId(int): void	Function used to set the seller ID I/P: <i>Id of the registered user who is selling</i> O/P: N/A
.getItemId(): int	Function used to get the ordered item ID I/P: N/A O/P: <i>Id of the order's specified item</i>
.setItemId(int): void	Function used to set ordered item ID I/P: <i>Id of the item to be ordered</i> O/P: N/A
.getPrice(): float	Function used to get the price of the item I/P: N/A O/P: <i>The total price of the order</i>
.setPrice(float): void	Function used to set item price I/P: <i>The total price associated with the order</i> O/P: N/A
.getDate(): String/timestamp	Function used to get the order date I/P: N/A O/P: <i>The specified timestamp the order occurred</i>
.setDate(String/timestamp): void	Function used to set order date I/P: <i>The timestamp of when the order occurred</i> O/P: N/A



## 11.3 Item

Represents a single item in the directory of items uploaded for sale.

Item	
.itemName: String	Item's name.
.itemId: int	Item's ID (key).
.sellerId: int	Id of registered user selling item.
.price: float	Id of registered user ordering item.
.details: String	Specific details of item.
.category: String	Category of item (book, clothes, etc.).
.itemPictures: Image array	User uploaded pictures of item. (3 max)
.setItemName(String): void	Set the name of the item <i>I/P: The name of the item</i> <i>O/P: N/A</i>
.getItemName(): String	Gets the name of the item <i>I/P: N/A</i> <i>O/P: The name of the item</i>
.setItemId(int): void	Sets the Id of the item <i>I/P: The Id number specified for the item</i> <i>O/P: N/A</i>
.getItemId(): int	Gets the Id of the item <i>I/P: N/A</i> <i>O/P: The Id number of the item</i>
.setSellerId(int): void	Associates the item with the registered user selling it. <i>I/P: The Id of the registered user selling the item</i> <i>O/P: N/A</i>
.getSellerId(): int	Gets the Id of the registered user selling the item. <i>I/P: N/A</i> <i>O/P: The Id of the registered user selling the item</i>
.setPrice(float): void	Sets the price of the item in dollars <i>I/P: The dollar amount</i> <i>O/P: N/A</i>
.getPrice(): float	Gets the price of the item in dollars <i>I/P: N/A</i> <i>O/P: The dollar amount of the item</i>
.setDetails(String): void	User entered details of the item added to item. <i>I/P: The details specified by the seller</i> <i>O/P: N/A</i>
.getDetails(): String	Gets the details of the item as specified by seller. <i>I/P: N/A</i> <i>O/P: The details specified by seller</i>
.updateDetails(): String	Seller can add details without removing previous. <i>I/P: The extra details to associate with the item.</i> <i>O/P: N/A</i>

.setCategory(String): void	Establishes which category the item belongs to. <i>I/P: The category the item belongs to</i> <i>O/P: N/A</i>
.getCategory(): String	Gets the category the item belongs to. <i>I/P: N/A</i> <i>O/P: The category the item belongs to</i>
.addPicture(Image, int): void	Adds a picture representing the item (max 3). <i>I/P: Image: The picture of the item</i> <i>int: The index (0-2) of the picture to add</i> <i>O/P: None</i>
.getPicture(int): Image	Returns one of the images associated with the item. <i>I/P: index (0-2) of one of the three pics to return</i> <i>O/P: The Image requested</i>
.updatePicture(Image, int): void	Replaces an image representing the item <i>I/P: Image: The picture to update with</i> <i>int: The index (0-2) of the picture to replace</i> <i>O/P: N/A</i>

## 11.4 Admin

Used to represent a single registered admin in the database.

Admin	
.username: String	Admin's username
.id: int	Admin's ID (key)
.emailAddress: String	Admin's email address
.password: String	A secure string that enables Admin to access their account.
.removeItem(int): void	Function used to remove an item <i>I/P: Id of item to be removed from ShopU</i> <i>O/P: N/A</i>
.removeUser(int): void	Function used to remove a user <i>I/P: Id of registered user to be removed from ShopU</i> <i>O/P: N/A</i>
.updateEmail(String): void	Function used to update Admin's email <i>I/P: The new email address of the admin</i> <i>O/P: N/A</i>
.updatePassword(String): void	Function used to update Admin's password <i>I/P: The new password for the admin sign in</i> <i>O/P: N/A</i>

## 11.5 Database

Class object that represents the database on the server, and functions to extract data from tables.

Database	
.getAllItems(): Item array	Used to get all items from DB. <i>I/P: N/A</i> <i>O/P: An array of all items</i>
.getItemsContaining(String, String): Item array	Used to get item from DB based on search arguments within a specific category. <i>I/P: String with search arguments</i> <i>String: Category to search item with</i> <i>O/P: An array of items that match search query</i>
.getAllUsers(): User array	Used to get all users from DB. <i>I/P: N/A</i> <i>O/P: An array of all users</i>
.getRegUser(int): RegisteredUser	Call a registered user's details from the DB to a RegisteredUser object. <i>I/P: Id of registered user</i> <i>O/P: RegisteredUser object of requested user</i>
.getAdmin(int): Admin	Call an admin's details from the DB to an Admin object. <i>I/P: Id of admin</i> <i>O/P: Admin object of requested admin</i>
.addItem(Item): void	Adds an item to the database <i>I/P: Item object to be added</i> <i>O/P: N/A</i>
.addRegUser(RegisteredUser): void	Adds a registered user to the database <i>I/P: RegisteredUser object to be added</i> <i>O/P: N/A</i>
.addAdmin(Admin): void	Adds an admin to the database <i>I/P: Admin object to be added</i> <i>O/P: N/A</i>
.updateItem(Item): void	Updates an item already in the database <i>I/P: Item object representing item to be updated</i> <i>O/P: N/A</i>
.updateRegUser(RegisteredUser): void	Updates a reg. user already in the database <i>I/P: RegisteredUser object representing user to be updated</i> <i>O/P: N/A</i>

<code>.updateAdmin(Admin): void</code>	<p>Updates an admin already in the database  <i>I/P: Admin object representing admin to be updated</i>  <i>O/P: N/A</i></p>
<code>.removeItem(int): void</code>	<p>Removes an item from the database  <i>I/P: Id of item to be removed</i>  <i>O/P: N/A</i></p>
<code>.removeRegUser(int): void</code>	<p>Removes a registered user from the database  <i>I/P: Id of registered user to be removed</i>  <i>O/P: N/A</i></p>
<code>.removeAdmin(int): void</code>	<p>Removes an admin from the database  <i>I/P: Id of admin to be removed</i>  <i>O/P: N/A</i></p>
<code>signin(int, int): RegisteredUser/Admin/null</code>	<p>User signin for registered user or admin.  <i>I/P: int: A code representing either registered user or admin (0 or 1)</i>  <i>Int: The Id of the user signing in</i>  <i>O/P: An object representing the registered user or admin who has successfully logged in (or null object if failed)</i></p>
<code>signout(int, int): boolean</code>	<p>User signout for registered user or admin.  <i>I/P: int: A code representing either registered user or admin (0 or 1)</i>  <i>Int: The Id of the user signing out</i>  <i>O/P: A boolean representing a successful signout (or not)</i></p>

## 12. Key Risks

**Skill Risks** - Several team members are new to using MySQL. To solve this, we are having the more experienced teammates work primarily in backend setup with the ability to assist in UI design and implementation questions.

**Schedule Risks** - Karen works three days a week. Pan works two days a week. Matt and Tyler intern elsewhere most days of the week. Valerio is busy. Although some team members are busy throughout the week, we are able to work out a schedule where most of the team can meet and work together.

**Technical Risks** - Still in the process of understanding how the APIs are going to work with the MVC architectural design. We have been researching and asking questions, and are getting to understand it more and flesh it out within our design.

**Teamwork Risks** - Occasionally some members are late or missing to group meetings, exhibit lack of participation. Will need to stay in contact with them more than others, encourage them to participate more, and potentially get higher-ups involved if problem persists.

**Legal/Content Risks** - At this point our team has no content risks because all logos and creative content used will be created by the team. A legal risk for our services section of the website is if a party posts fraudulent services to scam someone by getting their email or phone number.

## 13. Team Organization

- **Matt: Team Lead/Backend.** In charge of overall organization of the group and status updates of the software progress. Will divide the work fairly among the team and assisting with any software or teamwork problems that arise.
- **Tyler: CTO/Backend.** In charge of backend development, primarily databases and website efficiency. Will assist any technical issues that arise, frontend and backend.
- **Qihong: Frontend.** In charge of making the website comfortable and easy to use for the users. Make sure it is visibly pleasing.
- **Luis: Backend Lead/Frontend Support.** Will assist with any software difficulties that arise. Will work in backend development, primarily databases and website efficiency.
- **Pan: Frontend Lead.** In charge of guiding the frontend design, and making the website comfortable and easy to use for the users. Will make sure it is visibly pleasing.
- **Valerio: Frontend.** In charge of making the website comfortable and easy to use for the users. Will make sure it is visibly pleasing.

## 14. Attachments

7/9/2016

Gmail - CSC 648-848 Summer 2016: jQuery plugin



Luis Estrada <lierluis1@gmail.com>

---

### CSC 648-848 Summer 2016: jQuery plugin

2 messages

---

**Luis Estrada** <lierluis1@gmail.com>  
To: Marc Sosnick-Pérez <marcsosnick@me.com>

Fri, Jul 8, 2016 at 6:44 PM

Hi professor,

I found a jQuery plugin called DataTables: <https://datatables.net/index>

It provides in-table searches and pagination, but it doesn't negate the database search functionality that's required of us to implement for milestone 2.

Would it be ok to use it even though it isn't listed in the jQuery plugin registry ([plugins.jquery.com](http://plugins.jquery.com))?

-Luis

---

**Marc Sosnick** <marcsosnick@me.com>  
To: Luis Estrada <lierluis1@gmail.com>

Sat, Jul 9, 2016 at 2:15 PM

Yes, approved. Please be sure to include this email in your milestone document.

Marc  
[Quoted text hidden]