SWAMP

SW Engineering CSC648/848 Spring 2020

TEAM 04 - Milestone 01

March 5, 2020

Team Lead: Ben
Git Master: Weerachai
Front-End Lead: Dang
Back-End Lead: William
Front-End Dev: Onu
Back-End Dev: Kevin

Table of Contents

Executive Summary

Swamp is a media marketplace made by San Francisco State University students for other SFSU community members that offers trade and exchanging of media. Developed by gators, the creators of Swamp found that currently there is not a multimedia marketplace that caters only to SFSU students and faculty. Using modern software web application technology, our team is able to ensure quick, easy and reliable use of our product so that our users can have exclusive access to a reliable resource of media assets for academic or extracurricular usage.

Swamp offers a fluid user experience that is user-friendly for community members that may be familiar or unfamiliar with web application usage, while also offering features that tailor to needs of the student body. Our search system offers a feature allowing for students to filter listings posted by area of study. Upon entering the site, users are able to browse the site's marketplace. Once an unregistered user commits to a transaction, they will be prompted to register ensuring that the buying and selling feature is exclusive to registered users. Admin acts as a filter for content by approving or denying user listings so that all visible content is maintained to be appropriate.

Swamp's team is composed of upper division San Francisco State University computer science students and operates out of San Francisco. Each team member adds a unique perspective to the Swamp product while being able to effectively work as a team in order to create a product that delivers an efficient exchange of media between SFSU community members. Since Swamp is created by students, our team is able to shape the product toward the needs of the demographic we are serving.

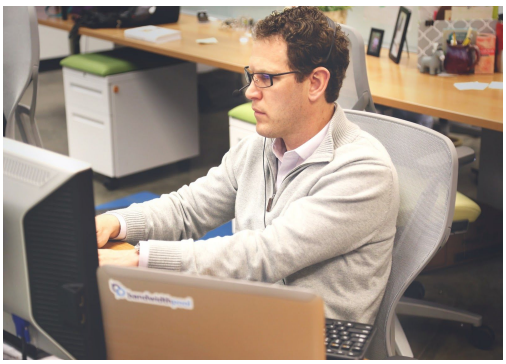
Personas

Registered - Emma

- She wants to use a website that can provide easy usage and friendly user interface
- She wants to help out students who are selling media with Swamp
- She wants to sell her media in the future

Unregistered - Robert

- He wants an easy search engine to use
- Website should not have too much confusion to use for him
- He needs less clicks to get what he wants



Admin - William

- Experienced programmer
- Can ban suspicious behavior
- Works part time on campus
- Knows school posting policy

Use Cases

Registered User -

Emma, a sophisticated student from San Francisco State, used to only buy required books from popular sites such as Amazon, Ebay, etc. These sites always sell digital book content overpriced as they do not cater their content toward students. Since being introduced to what Swamp's Services offers, she learned about how reliable and easy-to-use Swamp is. Now, when searching for a required digital content, she is able to find e-books sold at noncommercial prices (cheaper than our competitors). Not only is she able to easily receive assigned homework from her professors, but she is also able to download other types of digital media.

Unregistered User -

Robert, who works in construction, has little knowledge on how to use a website. He wants to go back to school. He realises that he needs to buy media content required by a class before the semester starts. He hears about Swamp Services from a professor. Swamp provides him with an easy browsing engine that allows him to find class content posted by his professors.

Admin User -

William is an experienced programmer who monitors Swamp's services. If the admin sees or is notified about inappropriate content he can either ban a user account or delete user posted content. Admin is in charge of approving items to be able to sell on Swamp. Every item has to have approval from administrators like William to be able to sell on Swamp.

Main data items and entities

Registered user	Users that own an account and are able to log in	Able to buy & sell products
Unregistered user	Users that do not own an account and but are able to registar	Only able to browse all listings
Item listing	Listed item that can be an image, audio, video, or document file	Entity to be exchanged between users through the marketplace
Metadata	File name, size, type, author, item description, and date created	Used for keeping track of items as well as for filter/ searching purposes
Admin	Administrative account that acts as a filter for listings	Approve or reject listings to ensure appropriate marketplace; ban users
Messaging	Users, upon checkout of listings, can message the owner in order to complete payment	Allows for ease of purchasing

Functional Requirements

Unregistered users **shall**:

- Be able to search for Swamp data
- Be able to filter their searches
- Be able to register
- Be able to add item from cart
- Be able to view from shopping

Registered users **shall**:

- Be able to logout
- Be able to login
- Be able to edit password
- Be able to post media for sale
- Be able to edit existing listing
- Be able to view all listings, requests, approvals
- Be able to request to purchase
- Be able to approve purchase request
- Be able to deny purchase request
- Be able to remove listing
- Be able to report listing

Administrative users **shall**:

- Be able to login
- Be able to logout
- Be able to approve posting
- Be able to delete posting
- Be able to view user reports
- Be able to ban user

Non-functional requirements

- 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).

- Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
- Selected application functions must render well on mobile devices
- Data shall be stored in the team's chosen database technology on the team's deployment server.
- Full resolution free media shall be downloadable directly, and full resolution media for selling shall be obtained after contacting the seller/owner
- No more than 50 concurrent users shall be accessing the application at any time
- Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
- The language used shall be English (no localization needed)
- Application shall be very easy to use and intuitive.
- Google analytics shall be used
- No e-mail clients shall be allowed
- Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
- Site security: basic best practices shall be applied (as covered in the class) for main data items
- Media formats shall be standard as used in the market today
- Media material shall be either free or for sale, as determined by media owner
- Each media material shall have its license info as one of the following: a) free use and modification; b) free but only allowed for SFSU related projects; c) for sale
- Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
- The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

Competitive Analysis

Competitive Analysis	Amazon	Craigslist	Swamp
Search	++	-	++
Support	+	+	+
Mobile	+	-	+

User Friendly	++	-	+
SFSU students	-	-	++

System architecture and technologies used

Server Host	Amazon Web Services
Operating System	Ubuntu v18.04
Database	MySQL v14.14
Web Server	Amazon EC2
Server-side Language	NodeJS v12.14.1
Front-end Framework	ReactJS v16.9.0
Back-end Framework	Express v4.16.1

Team and roles

Team Lead: Ben

Git Master: Weerachai

Front-End Lead: Dang

Back-End Lead: William

Front-End Dev: Onu

Back-End Dev: Kevin

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 are working on learning and practicing - IN PROGRESS
- Team lead ensured that all team members read the final M1 and agree/understand it before submission - DONE
- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) - DONE