San Francisco State University

SW Engineering CSC648/848 Spring 2020

Snapster

TEAM 06

Bakulia Kurmant, Team Lead. Email: bkurmant@mail.sfsu.edu

Akhil Gandu, Back End Software Engineer and GitHub Master

Chris Eckhardt, Back End Lead Software Engineer

Elliot Yardley, Front End Lead Software Engineer

Thomas Yu, Front End Software Engineer

Avery Chen, Back End Software Engineer

Milestone 4

May 18th, 2020

Date Submitted	Date Revised
May 18th, 2020	

Table of Contents

Title Page

Table of Contents

Product Summary ……..……..…….…....3

Usability Test Plan..……....…….....4

Test Objectives…….....4

Test Background and Setup……4

Usability Task Description…….....5

QA Test Plan …….……...……..…..…..…7

Test Objectives…….…..7

HW and SW Setup…….7

Feature to be Tested……...7

QA Test Plan……..8

Code Review…….…….....9

Self-Check on Best Practices for Security……..…….....….11

Self-Check: Adherence to Original Non-Functional Specs - Performed by Team Leads ……....12

1. Product summary

Snapster is a web platform created exclusively for students and faculty members at San Francisco State University, where our customers can buy, sell and share digital media with ease and transparency, particularly, we offer the ability to post items for sale or free of charge only between students and faculty at San Francisco State University, as well as for purchase or downloads for free.

Snapster is created by motivated students of six at San Francisco State University, who want to build an easy to use and versatile online marketplace to serve the needs of the Gator community, and improve student cooperation across academic disciplines.

Our mission is to provide reliable, secure, and practical functionality to our fellow students and faculty as follow:

- Search and browse digital media.
- Search and browse by digital media categories.
- Search and browse by filtering digital media posted free or for sale.
- Search and browse by San Francisco State University Course name.
- See digital media metadata.
- Register with their San Francisco State University email, username and password.
- Login with their email or username and password
- Sell, buy or share digital media as a verified San Francisco State University member.
- Post digital media for sale/download
- Message the seller in order to buy the digital media.
- View their dashboards for transactions.
- Users are verified by their San Francisco State University emails and their data is protected by the administrator.

Please visit our web platform and start posting and downloading the digital media you need for school projects, homework or research at www.snapsster.com.

2. Usability test plan

To develop a usable product we applied User Centered Design and needed an usability testing to verify usability. The test will address several key questions the front-end team needs answers to for the next iteration. Failing to answer these questions now increases the risk of developing the unusable product for our users.

Test objectives

Our main product is search for digital media. First we will be testing the effectiveness of the search bar if it provides accurate and complete digital media results the users expect. Second, we will be testing the efficiency of the search bar which provides the resources such as average time to complete the task, number of clicks, number of screens expended in relation to the accuracy and completeness with which users achieve goals. Third, we will be testing users satisfaction in order to analyze the comfort and acceptability of use.

Test background and setup

The search function on www.snapsster.com/search will be tested. The purpose is to have the search bar to be used easily and intuitively. When the users go to the main page of the web platform they shall be able to type for digital media they want to search and browse and the search function shall return results the users expect. The intended users are San Francisco State University students, who need digital media for their homeworks and researches; students who major in art, journalism and photography who want to share their work with the their community, and faculty members who want to share their own work or recommendation on digital media to their students in order to succeed in their classes. The following questions will be tested and measured: Do the users find the search bar easy and intuitive to use? Do they find categories, media types and license types easy to use? Are they happy with the search results? Do they find what they look for? Do the users trust the service?

Usability Task description

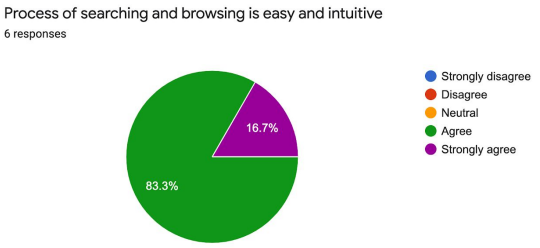
Task 1.	Goal: Getting idea about the website Snapsster
Steps	- Go the website www.snapsster.com - Spend 30 second max to browse the website and find the search bar
Task 2.	Goal: To what extent searching and browsing the digital media is easy by categories?
Steps	- Search for an image in ART CLASS category
Task 3.	Goal: Searching and sorting features.
Steps	- Use the search bar to find videos for COMPUTER SCIENCE class that are FREE for download
Task 4.	Goal: Single digital view page based on the search results.
Steps	- Search for any digital media you need - Open the digital media to view more information about it
Task 5.	Goal: Do users easily find the registration and login page.
Steps	- Register or login using their SFSU email account.
Task 6	Goal: Do users find it easy to contact the seller
Steps	- Contact the author to download a digital media

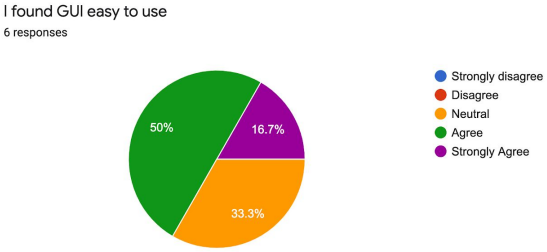
To measure effectiveness of usability metrics, search function is chosen for the intended users with average computing skills. Two options to evaluate for graphic user interface functionality. First, search digital media on the search bar or browse all digital media.

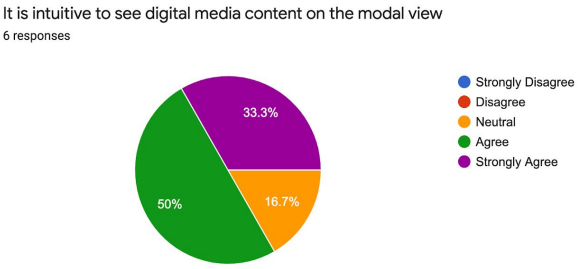
In order to measure efficiency in usability metrics, average time to complete the tasks, number of clicks the users make searching and browsing for digital media they want, content and design on the web pages will be measured by numbers of screens and pages of instructions.

To measure users' satisfaction in usability metrics, a group of six were asked to perform tasks and surveyed anonymously on a Likert questionnaire (created with google forms) how they "feel" about usability.

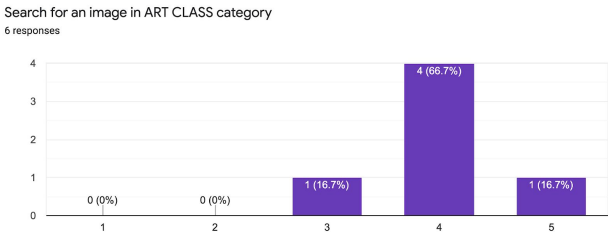
First, the users were asked to find the search bar and browse the digital media posted on the web page. See the charts below.

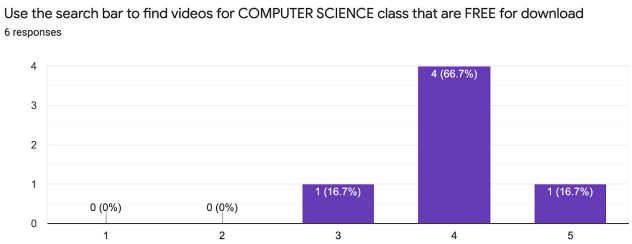






Second, the users were asked to search for specified digital media and rate between 1-5:
1 being Strongly Disagree to 5 being Strongly Agree.





3. QA test plan

The quality assurance test plan has been created by our team to ensure that the program performs according to specifications. It includes test objectives, hardware and software setup, features to be tested and testing approach. This will identify what the test deliverables will be and what is deemed in and out of scope.

Test objectives

Team members Akhil and Avery are responsible for testing the product and ensuring it meets their needs. The testers are both the consumer and the tester in this project. Must have functionality requirements listed in priority one and delivery date are important in this project.

HW and SW setup

For black box testing strategies testing will be initiated on the basis of requirements specifications document, and will be done by trial and error ways and methods.White box testing will be done by testing the logical database, searching algorithms and internal structure of the code.

We tested both on Windows and MAC operating systems using Chrome, Firebox and Safari browsers as well as mobile browsers. Tests were performed on web server www.snapsster.com

Feature to be tested

- Server
- Search function
- Upload Media
- Single media view
- Registration
- Messaging
- Database logical
- Pagination

QA Test Plan Table

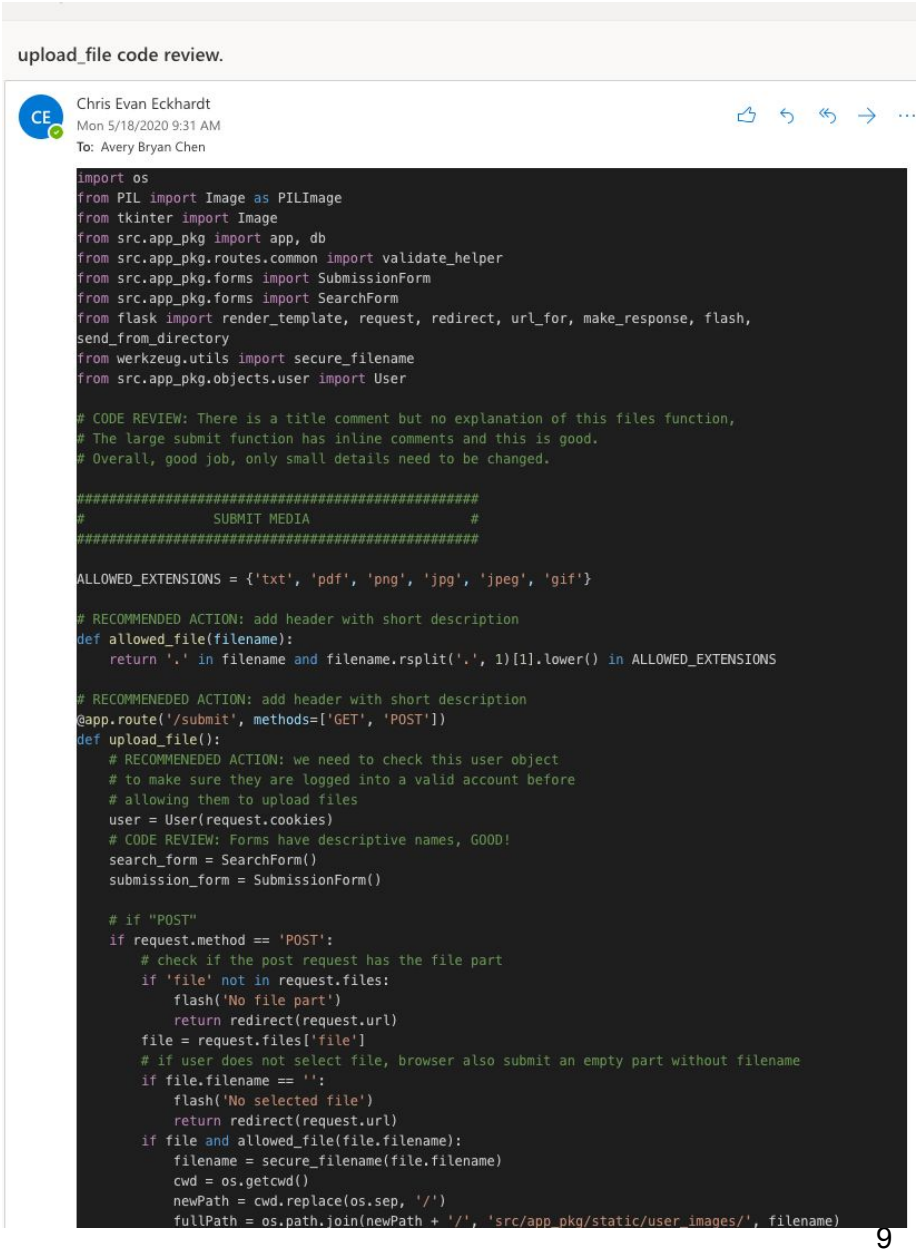
No	Description	Test input	Expected output	Pass/Fail
1	Test % like in search for name field	Type “sponge bob” in search field	Get 5 results, all have “sponge bob” in name field	Pass
2	Web page on www.snapsster.com can be opened on major browsers including mobile	Open the web site from Chrome, Firefox, Safari and Mobile Browser	The web page is responsive in all browsers	Pass
3	Users can search and browse digital media by categories, media types and license types	Type Digital media item in search bar filtering by “image”, in “biology” class with “free” of charge	The search results returns all digital media of type “image” under “biology” class category with “free” license type.	Pass
4	Users can upload media for free of charge or sale	On the nav bar top right press on SUBMIT MEDIA to upload a media for free and sale	SUBMIT MEDIA button guides users to upload digital media with tille, media type, category, license type, description and thumbnail image.	Fail *Is being fixed
5	Users can register only with San Francisco State University email.	Register with username, sfsu email and password	Users are able to register with sfsu email	Pass
6	Users can contact the seller to obtain the digital media	Contact the seller in order to download digital media	Contact seller button appears, where the user send a message to the author of the digital media is posted for sale	Pass
7	Queries from the database must meet users needs and be persistent.	Login to your account and search for digital media	Registered users successfully login and search results meet users needs.	Pass
8	Search results can return up to 12 items per page.	Type digital media you want to download	Search results show up 12 digital media matched with with the name field	Pass

4. Code Review:

As a programming language we have chosen Python. To format code correctly we referenced to https://google.github.io/styleguide/pyguide.html#Naming. The back end team lead reviews the code styling and keeps it consistent throughout the development of the application. Function headers are required for all function definitions. Inline comments are only required for large or complex functions. Every file has a header comment describing its contents. No code is pushed to the master github branch without thorough testing by a minimum of two other teammates. No code can be pushed to the master branch by anyone except the team's Github master, Akhil.

As part of the QA test plan, the upload digital media feature was chosen to show an example of the code under review with the back end team members.

Please see the images.



upload_file code review.

Chris Evan Eckhardt
Mon 5/18/2020 9:31 AM
To: Avery Bryan Chen

```
import os
from PIL import Image as PILImage
from tkinter import Image
from src.app_pkg import app, db
from src.app_pkg.routes.common import validate_helper
from src.app_pkg.forms import SubmissionForm
from src.app_pkg.forms import SearchForm
from flask import render_template, request, redirect, url_for, make_response, flash,
send_from_directory
from werkzeug.utils import secure_filename
from src.app_pkg.objects.user import User

# CODE REVIEW: There is a title comment but no explanation of this files function,
# The large submit function has inline comments and this is good.
# Overall, good job, only small details need to be changed.

##############################################
#                SUBMIT MEDIA                #
##############################################

ALLOWED_EXTENSIONS = {'txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'}

# RECOMMENDED ACTION: add header with short description
def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# RECOMMENEDED ACTION: add header with short description
@app.route('/submit', methods=['GET', 'POST'])
def upload_file():
    # RECOMMENEDED ACTION: we need to check this user object
    # to make sure they are logged into a valid account before
    # allowing them to upload files
    user = User(request.cookies)
    # CODE REVIEW: Forms have descriptive names, GOOD!
    search_form = SearchForm()
    submission_form = SubmissionForm()

    # if "POST"
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']
        # if user does not select file, browser also submit an empty part without filename
        if file.filename == '':
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            cwd = os.getcwd()
            newPath = cwd.replace(os.sep, '/')
            fullPath = os.path.join(newPath + '/', 'src/app_pkg/static/user_images/', filename)
```

upload_file code review.

```
        # if user does not select file, browser also submit an empty part without filename
        if file.filename == '':
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            cwd = os.getcwd()
            newPath = cwd.replace(os.sep, '/')
            fullPath = os.path.join(newPath + '/', 'src/app_pkg/static/user_images/', filename)
            file.save(fullPath)
            user_images = os.path.join(newPath + '/', 'src/app_pkg/static/')

            #makes thumbnail and saves it to thumbnails folder
            f = PILImage.open(fullPath)
            f.thumbnail((200, 200))
            f.save(user_images + 'thumbnails/t_' + filename)
            print("thumbnail saved")

            # query db params, add approval variable
            session_token = request.cookies.get('token')
            name = request.form['filename']
            desc = request.form['description']


            license_val = request.form['license_field']
            print("Value", license_val)
            price = request.form['price'] if license_val == "2" else 0.00


            cat = request.form['category']
            media = request.form['media_type']
            filepath = 'user_images/' + filename
            thumbpath = 'thumbnails/t_' + filename

            # RECOMMENEDED ACTION: print statements should be left for diagnostic purposes only,
            # if they are useful for debug and you wish to keep ityou should comment it out.
            print(name, " ", desc, " ", price, " ", cat, " ", media, " ", filepath, " ", thumbpath, "
", session_token)\

            db.upload_file(user.user_id, name, desc, filepath, thumbpath, cat, media, price,
session_token)
            submission_form.filename.default = filename
            submission_form.description.default = desc
            submission_form.price.default = price
            submission_form.category.default = cat
            submission_form.media_type.default = media
            submission_form.process()

            return redirect(url_for('search'))

    # else, (if "GET")
    return render_template('search.html', search_form=search_form, submission_form=submission_form)

# RECOMMENEDED ACTION: add a short header comment with description of function
@app.route('/uploads/<filename>')
def uploaded_file(filename):
    return send_from_directory(app.config['STATIC_PATH'] + 'user_images/', filename)
```

5. Self-check on best practices for security

Assets are protected

User privacy and data must be safe and well protected. The following assets are protected: the user database, messages and digital media they post for sale.

Threats

1. Unauthorized user, who is not a San Francisco State University student or member, gains access to confidential data.
2. Unauthorized users are able to read and modify messages between arbitrary parties.
3. Unauthorized users deleting or altering media that they do not own.

Threat Control

1. Require users to authenticate themselves by registering and signing in.
2. We require each user to have a session token, and the identifiers used for messages are never directly exposed on the client side.
3. We are verifying the users session token each time they attempt to modify a digital media item, and we implemented access controls to prevent them from being able to modify items that they do not own.

PW Encryption in the DB

Password is hashed and salted using the bcrypt library for python.

Input data validation

- We are validating the user's email address to make sure it ends in sfsu.edu.
- At least 8 characters including numbers are required for password.
- We are restricting the size of the search query to 40 alphanumeric characters.
- We are restricting the size of the title of the uploaded digital media items to 40 characters.
- We are restricting the size of the description of the uploaded digital media to 120 characters.

6. Self-check: Adherence to original 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 two major browsers **DONE**
3. Selected application functions must render well on mobile devices. **DONE**
4. Data shall be stored in the team's chosen database technology on the team's deployment server. **DONE**
5. Full resolution free media shall be downloadable directly, and full resolution media for selling shall be obtained after contacting the seller/owner. **ON TRACK**
6. No more than 50 concurrent users shall be accessing the application at any time. **DONE**
7. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **DONE**
8. The language used shall be English (no localization needed). **DONE**
9. Application shall be very easy to use and intuitive. **DONE**
10. Google analytics shall be used. **ON TRACK**
11. No emailing clients shall be allowed. **DONE**
12. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **DONE**

13. Site security: basic best practices shall be applied (as covered in the class) for main data items **DONE**
14. Media formats shall be standard as used in the market today. **ON TRACK**
15. Media material shall be either free or for sale, as determined by the media owner. **DONE**
16. 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. **DONE**
17. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development. **ON TRACK**
18. 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). **DONE**