Observation and User Accounts and Socket.IO

No screens

Prof. Lydia Chilton
Adv Web Design Studio
4 October 2019
Primary Objective:

Make software that impacts people
Not a portfolio project... **an impact** project
You cannot make impact with technology alone.

You have to meet people,
learn their needs,
and help them achieve their goals.
Example of an impact project from Adv Web Design Studio Fall 2018

There are 11,972 empty walls in NYC.
This is Susie. She owns a cafe—and has plenty of wall space.

This is Zak. He’s an artist and needs somewhere to hang and sell his art.
You just completed a full first iteration of an impact project
Welcome to Potluck House.

Cook

Click on Cooks to add your dish to the list for this week's potluck.

Eaters

Click on Eaters if you plan on attending to vote for cooks' dishes.

Suggestions

Click on Suggestions to suggest a dish or if you have dietary restrictions.
Full Iteration:

- Brainstorming user groups on campus
- Technical Prototype
- Graphic Design
- User testing
- Implementation
- Impact
- Pitch

But that was just a warm-up
Now, we’re going to find real user needs through observation and interviews. Rather than assuming we know what the problems are, we are going to look closer at situations before deciding what is needed.
Find a group on campus to observe

• You already brainstormed 10 groups.
• Pick one that might have interesting technology needs.
• Hopefully, you already have insight into this group.
• Observe them (two sessions) doing their work.
• And be able to list 5 examples of:
  • activities—goal-directed actions, activities, and processes
  • environments—personal or shared workspaces or common areas
  • interactions—between people and objects
  • objects—things people have in their environment and use in their activities
  • users—the people you’re observing
Let’s do it now for this class. For each category, what are 5 things we do that involve technology:

- **activities**—goal-directed actions, activities, and processes
- **environments**—personal or shared workspaces or common areas
- **interactions**—between people and objects
- **objects**—things people have in their environment and use in their activities
- **users**—the people you’re observing

Does anything stick out as a critical incident?
4 Typical problem technology solves

- **Marketplaces**
  - OkCupid
  - ?

- **Communication**
  - Slack
  - ?

- **Databases**
  - IMDB
  - ?

- **Workflows**
  - Turbo Tax
  - ?
What might be some of the technology needs for Advanced Web Design Studio?
Assignment Part 1: Do Observation in Pairs

• Read article about observation
• Find a partner.
  • You will both identify a group to observe
  • And both go observer the group and write up your observations
• Read the article on Observation
• Do this process for both observations
  • Work as a team, write it up separately.
• Take pictures
• Be prepared to present interesting findings (in PPT) in small groups.
Assignment Part 2: A Chat App

• Technical Prototype of a chat application
• Must have user login with WTForms and Flask_login
• Must have real time message broadcasting with Socket.IO
• There is example code.
• Get the code to run locally
• You have to make up some random use case for it.
  • Add ONE feature for this use case.
  • Put a tiny bit of UI work into it.
• Submit a video of you using it in two browsers side-by-side
User Accounts
Basic elements of Login

**Front end:**
User actions needed

- Register
- New Users
- Login
- Logout
- Using app

**Back end:**
Stuff needed to support user actions

- Database of users
- Password hashing

**Correct**
- Bad: User already exists
- Good: created. Now what?

**Incorrect**
- Where to send the user next

**Where to send the user next**
- Know they are signed in
What to implement:

**Register New Users**
- A form
- Feedback

**Login**
- A form
- Feedback

**Logout**
- A button

**Using app**
- Reminder you are logged in

**Front end: User interface**

**Back end: Database interaction**
- Validation
- Validation
- nothing
What to implement:

**Front end: User interface**
- Register New Users
- Login
- Logout

**Back end: Database interaction**
- Validation

**Front end: User interface**
- A form
- feedback

**Back end: Database interaction**
- Validation

**Using app**
- Reminder you are logged in
- nothing

**WTForms**

**Flask_login**
- nothing
Flask_login:
log users in, log users out, know who they are

Add flask_login:

```python
from flask_login import LoginManager

login = LoginManager(app)
```

@login.user_loader
def load_user(id):
    return User.query.get(int(id))

Use flask_login:

```python
user = Users.query.get(id)

login_user(user)
```

```html
< div >
    Hi, {{ current_user.username }}!
</ div >
```
WTForms
Define forms and their validation

**Back-end:**
Define Form class

**Front end:**
render form

code sample:

```python
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, BooleanField,
from wtforms.validators import ValidationError, DataRequired,
from app.models import User

class LoginForm(FlaskForm):
    username = StringField('Username', validators=[DataRequired()])
    password = PasswordField('Password', validators=[DataRequired()])
    remember_me = BooleanField('Remember Me')
    submit = SubmitField('Sign In')
```
Socket.IO
Getting new data: Pull model vs. Push model

How do users get new data from the IMDB server?

Pull model – driven by user clicks

How do users get new data from the GMail server?

Pull model – Driven by a timer on the client side

How do users get new data from the Twitter server?

Push model – Driven by updates on the server

For the server to push data, we need more than HTTP. We need WebSockets. SocketIO implements WebSockets.
Client **Pull** Info from the server (by asking)

**Clients**
- Serina
  - Post this T-shirt I want to sell

**Server**
- Post this T-shirt I want to sell

**Priya**
- ?????
- NOTHING.
Client **Pull** Info from the server (by asking)

**Clients**

Serina: Post this T-shirt I want to sell

Grant: Any new t-shirts? (refresh the page)

“BUY IT!!!”

**Server**

Post this T-shirt I want to sell
Can we simplify this?

Clients

Serina: Post this T-shirt I want to sell

Grant: Any new t-shirts? (refresh the page)

“BUY IT!!!”

Server

Post this T-shirt I want to sell
Can we simplify this?

Serina: Post this T-shirt I want to sell

Julia: “BUY IT!!!”

Grant: “BUY IT!!!”

Chris: “BUY IT!!!”

Server

Post this T-shirt I want to sell
Socket.IO uses **pushes** new info to users

**Clients**

Serina: Post this T-shirt I want to sell

Julia: “BUY IT!!!”

Grant: “BUY IT!!!”

**Server**

Post this T-shirt I want to sell

Chris: “BUY IT!!!”
SocketIO is a framework to send and receive messages

Clients

Serina
Grant
Chris:

$(document).ready(function(){
  var socket = io.connect('http://localhost:5000/');
  socket.on('connect', function(){
    console.log("User has connected")
  });

  $('#sendButton').on('click', function(){
    var myMessage = $('#myMessage').val();
    socket.send(myMessage)
  })
})

Server

from flask_socketio import SocketIO, send
socketio = SocketIO(app)

@socketio.on('message')
def handleMessage(msg):
  send(msg, broadcast = True)
Step 0.

• Download the example code with User Accounts and Socket.IO
  • Get it to run on your machine ASAP.
  • You will have to pip install libraries, and do some config stuff.
  • No Digital Ocean (yet!)
If you feel like your brain is split in half...

That’s okay!

Engineering! Yay!

Humanity! Double yay!
RIGHT NOW

• Identify you teammate, and talk about what groups you each want to observe

• Reply to the piazza post called “My Observation group is”
  • Your names
  • What group each of you would like to observe this week.

• Only one of you needs to reply to the post. (this is how we record participation for today)