

Project Goal

Your team is to build a system that puts together different existing web components in an application that provides a quality user interface to the joined components. (Such applications are often called mashups.) The project is meant to be relatively open-ended and give your team the freedom to make many decisions regarding the form of the project itself. A key focus of this project will be the process, including collecting and using user feedback.

Here are the basic requirements for the system itself:

- You should create a web-based application. That is, the application should have its primary mode of interaction being through a browser interface. It must run, at minimum, in Chrome; you may also let it run in other browsers.
- Your system should incorporate at least 3 existing web services within it.
- Your application should be based on HTML and should not require any extensions to run on a web browser

Team Allocations

Team allocations can be found here:

<https://docs.google.com/spreadsheets/d/1ibW2IZGWejQUcl8Dz0zydGmnuHcX0Rww2mcHZCflQaU/edit?usp=sharing>

Project Organization and Deliverables

The following are deadlines for the various deliverables and other important dates:

- **Friday, November 18:** Project Assigned
- **Tuesday, November 28:** Initial prototype – **Deliverable 1**
- **November 27-December 1:** Conduct Formative User Study
- **Sunday, December 3:** User Study report – **Deliverable 2**
- Final Class days or final exam times: Final Project presentations – **Deliverable 3**. Here is the final exam schedule:

Sections	Date	Time
501-503	December 8	12:30 – 2:30 pm
504-506	December 12	1 – 3 pm

- **Tuesday, December 14:** Final Code, Team Retrospect Report, and Individual Report (if necessary) due – **Deliverable 4**

Note that the due date for Initial Prototype is November 28, while the user study starts from November 27. This is done so that the teams who are done with the prototype can start the study early. Also, some of the requirements for the user study need to be done before the user study is actually conducted! Please review the entire scope of the project to ensure that you are on track to meet the various requirements.

Initial Prototype

At the first deliverable stage, your team should have determined the goal for your project and developed an initial prototype. This prototype should be sufficient for conducting an initial user study. It does not need to be fully functional: there can be placeholder images, queries that do not work, missing text/information/etc. If necessary, you may even allow manual intervention:

you could have the user indicate what button got clicked, for instance, and then have one of your team members manually move to a new screen. However, there should be at minimum a sufficient user interface framework in place to allow you to present the system to others and get feedback during the stage. Here is what is required at the deadline:

- Application Name: You should choose a compelling name for your project that clearly conveys the goal of the project and what it does.
- Project Description: Give a description of the application. This should be 1-2 pages (single spaced) giving:
 - The goal of the project: what do you want this application to be able to do?
 - Planned Scope. Include both fallback and stretch goals. The scope should describe the range of functionality you expect to provide in your application. Fallback goals should indicate which things will be dropped first if your project is unable to achieve its planned scope. Stretch goals should indicate features that you believe are unlikely but possible to achieve, if you have sufficient time.
 - You should set the project up and develop it so that it is possible to move to your fallback and stretch goals if possible – i.e. that you will be “guaranteed” to meet some minimal level, and that the additional features can be incorporated without starting from scratch.
 - What existing web services you will be using or combining together in your application. Why did you pick these?
 - Background story: give a short “story” describing a situation in which a user would want to use your application: what is the user’s motivation, what experience do you expect the user to have coming in, what should the user’s impression/reaction be as they use it, what do they hope to gain from it by using it, etc.
 - You may give more than one such story, especially if you want people to use the application from different viewpoints
 - Development challenges: What development challenges do you foresee with respect to the project? This could include learning frameworks or services you are unfamiliar with, difficulty in getting authentication and similar privileges, etc.
 - Organization plan: Who will work on what during the project. You should have at least an initial breakdown of task responsibilities clearly delineated. You may (but are not required to) use a backlog and assign backlog items to individuals, for instance.
- Interface sketch. You should prepare a sketch/mock-up of your user interface. This should demonstrate:
 - The sketch may be handwritten or electronic – the idea is to design what you expect the interface to look like before actually building it.
 - The planned visual layout of the interface on a web page, with notes regarding color, images, etc.
 - Notes about the functionality, including what various controls might be provided and what they might do.
 - If your application will have multiple levels, you might need to have sketches of several screens.

- You may use any tools you wish to prepare this (including electronic ones or physical ones). Your submission needs to be in a PDF file, however. You may wish to take pictures or scan handwritten or physical diagrams, for instance.
- Code Submission. You should submit your source code for your prototype. The prototype is not expected to be fully functional (and in fact might be totally non-functional), but should be sufficient for you to perform the user study described below. Note that the description and interface sketch should probably be completed before the code is developed.

User Study

The second stage of the deliverable is conducting a user study. The goal of this study should be to evaluate the interface that you have begun developing in the prototype (if you have to, you may need to resort to using the sketch for parts of the study). The study should involve the user trying to perform a task or multiple tasks and an evaluation of that user's experience. There are three stages to this: designing the study, performing the study, and analyzing the study.

Documentation will need to be provided for each of these stages. Your User Study stage should include the following from each:

- A description of the user study design. Before you conduct your user study, your team should put together a description of exactly how the user study will work (and then use this in the actual study). The expectation is that you will do this using an interview process. **Your study must include the collection of qualitative data**, though you may also collect quantitative data; quantitative data is less likely to be useful at this stage. Try to avoid giving "on a scale of 1 to 5" type questions. You should submit your description at the second deliverable point. This should include:
 - How many participants and how long do you expect the evaluation to take? In determining times, be realistic about the time available to conduct these studies, noting that you will also have to participate in others' studies.
 - Where and when will you conduct the studies (in lab, outside of lab, etc.)? The expectations are that most will be in lab, but you may propose additional locations/times.
 - What pre-study questions will you ask the user?
 - What tasks will you have the user do? How much time will they take?
 - What will you measure from the user during the tasks? How will this information be collected?
 - What after-study questions will you ask the user?
- Conducting the user study. You are expected to conduct user studies during lab time between April 19 and April 25, using other members of the class as your users. You may run studies outside of lab time, if you wish, and may use people from outside of the class.
 - You must conduct studies with at least 3 users. You are encouraged to do more than 3 such studies.
 - Every person must participate in a user study for at least 1 other team. If you are unable to find a team to do a user study for, see the TA, who can try to help match you with a team needing a user.
- User study deliverable:

- For the deliverable, you are to include the raw data that you collected from the users. If you conducted interviews, include the notes taken. If you used surveys, include the survey results. Make the results easily accessible: e.g. a document or basic spreadsheet, even if you keep them in some more complex manner. If you perform video or audio recording, you do not need to include that, but should note that this was collected.
- In a **separate** document, give the names of the people from the class who participated in the user study. Normally, this information should not be kept as part of a user study; we will be collecting it in this case for purposes of ensuring participation by class members, only. The names should NOT be associated with the study data that you report/store.
- Statement of Revision: You are to prepare a document that analyzes the results of your user study and describes how you will change your interface as a result. Your statement of revision should include two separate parts:
 - An analysis of the qualitative (and, possibly the quantitative) data you collected. What problems were exposed in your design through the user study? What is working well? How do you know?
 - You are encouraged to use a basic coding and categorization approach to the data you collected, to draw out some of these conclusions.
 - Pay attention to issues such as mechanics, consistency, feedback, differing conceptual models, etc.
 - A description of planned changes. Describe how you will change your application as a result of this study. The changes should directly reflect the lessons learned from the data analysis. You may include additional sketches if desired.

Final Project (code)

The final project code is due the night before the final presentations. You should save a copy of your code from the github repository that you submit in a zip file to ecampus. Since your project is a web application, the expectation is that it will be running online through the end of the semester, to allow for grading. A team contact should be available to answer questions, and possibly run demos, through the end of final exams.

Final Project Presentation

You will be required to give a short (4-5 minute) demonstration/presentation of your application as a team, in front of the class.

- This presentation will be peer-evaluated, and all students are expected to attend the presentations and participate in the peer grading.
- All team members are expected to participate in the presentation in some way. Not everyone has to speak, but no one should just be standing around.
- The expectation is that we will have a single web browser running on the classroom machine, and your application should run in that browser window. You should not expect to use powerpoint or other materials in your presentation, but should be ready to provide a good verbal description of the application and demonstrate its use.
- It is expected that there will be two possible times for project presentations: the final day of regularly scheduled class, and the final exam time. Those teams wishing to finish their

project early can present at the earlier time, those wishing to have longer (but work on it into finals week) can present at the later time. We will conduct surveys to determine the exact times for each team to present.

- The presentation will be short, so you should practice and time the process as a team to ensure you can demonstrate your work.

Team Retrospective Report

The team retrospective should detail the overall process, including what was and was not accomplished, the effectiveness of the user evaluation, who worked on which parts, and any documentation needed for the report. It should include contact information for someone (or multiple people) who will be available to answer email questions on short notice through the end of final exams, as well as whether that person would be available to demonstrate the software if needed. The team retrospective will also include multipliers for the team members, as in prior team assignments. If the team does not agree on multipliers, then individual reports will be filed, as before.

Further details about the format of the final team retrospective and the final project presentations (format, schedule, grading rubric) will be provided later.

Other project organization

- You should use github for managing your code. Ideally, each person working on the code should update the code each time a feature is added. There should be regular check-ins of code, and no items should be considered “complete” unless they have been checked into github!
 - Please give the TA and instructors access to github.
- You may wish to maintain a backlog/burndown chart, hold SCRUM meetings, etc. In fact, this is encouraged if you found that it helped organize your second project. However, doing so is not a requirement in this project.
- We will have only a couple of class meetings during the project. Thus, it is important to pay attention to piazza and to email for updates. Some of these (e.g. to sign up for a slot for final presentation) will include critical information for the project.

Application Goals

The application itself is meant to be a very open system – this is a chance for your team to come up with and put together a web application that you find interesting. There are numerous web-based services that are available to incorporate, including such things as Google maps, Twitter, Youtube, Doodle (polls), Accuweather/Weather Channel/Weather Underground/Yahoo Weather, Pinterest, ebay, Flickr (photo sharing), Facebook, reddit, last.fm (music), Amazon Product Advertising, Instagram, etc. A website that is good for browsing a wide variety of APIs is ProgrammableWeb.com. There is basically no restriction on what services you use in your application, but be sure you obey any usage guidelines from the web service itself.

Many of these services will have short tutorials and online help describing how to use the API. You will need to allow some time to explore the services that each one offers in order to understand what each one is capable of providing.

Note that depending on what you implement, you may need to have both a server and a client, as opposed to simply a client. You are not required to have a server piece to your project, though.

The following are some ideas to help spark ideas for a team project. They are intended as a way to give you some starting points, and are not meant to limit you or direct you in fixed directions. You are encouraged to develop a project that your team members find interesting.

- Combining an image repository that includes location information with a mapping service, so that it is easy to visualize where photos were taken and select them based on area.
- Creating an application that lets you identify artists of interest and find media they have produced on multiple platforms.
- Looking up company stock information along with recent news events about the company.
- Etc.

Grading

There will be a team grade assigned for the project, and like the first project, the team grade will be apportioned out to the individuals. The rough breakdown of team grade is:

Criterion	Points (%)
Clear and complete project description	5
Creation of an initial prototype sufficient for user testing purposes	10
Outline of user study process	5
Participating in user study and peer grading	5
Analysis of user study, statement of revision, and incorporation of user study into final design. Was the information collected from the user study incorporated into the final user design?	15
Quality of user interface. How well does the user interface follow basic principles of good user interface design?	15
Functionality of the system. Does the system work as it was intended?	25
Final presentation: did the team effectively demonstrate the system as a whole?	10
Quality and creativity of the system (additional quality)	5
Code style (naming, layout, etc.)	5
Total	100

Some portions of the grading will be accomplished by peer grading of your final project presentation. This includes the final presentation and the quality/creativity points that could be earned.