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. It is as if you have been given funding for an initial prototype of a new product. A key focus of this project will be the process, including collecting and using user feedback. Your product can include different languages behind the scenes, but the overall goal is a quality user interface that also includes an element of accessibility. The thought is that this will be a web page, accessible to anyone with a browser, but successful projects have built stand-alone apps (With prior approval.)

Here are the basic requirements for the system:
- You should create a web-based application. That is, the application should have its primary mode of interaction 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. The following are API's that may be used but do not count as any of the 3 required existing web services:
    - Google Map
    - Google Calendar
    - Anything IMDb
- Your application should not require any extensions to run on a web browser.
- You should include consideration for color blindness and one other area of accessibility in your design. For an introduction to Web Accessibility, see this link. This will also be addressed in the Project Proposal.
- Your final product will be presented during Finals. The presentation is a Business Casual presentation where every member should take approximately equal time presenting. The goals of this presentation are to show your product and the evolution of your product from the initial design to the current one, and to convince those funding you to continue. Due to the change in class structure, please expect details on the presentation to come later. This will not affect the project itself, just how you demonstrate it to the class.
- A bonus will be given if the basics of this project are met and you design your application to be mobile friendly.

Team Assignments:
Teams have been assigned and will be posted prior to this being released. Make every effort to contact your team and set up appropriate lines of communications. There are four reports due with this project. Teams should work together on the reports and everyone is required to read and sign off on the report prior to it being turned in. Reports are a team assignment and only one copy needs to be turned in.

PROJECT ORGANIZATION

This project will be approached by using an *Agile* development environment. Due to team size, and even more significantly, due to time constraints, we will not be able to follow a "real" agile approach, but we will try to capture some of the main themes from a SCRUM-style development process. This project contains no planned outages. But then this semester contained no planned viruses. Even so, time management will be crucial to your success. Requirement changes from your initial design will come from the user studies that you provide. No external requirement changes are planned for this project.

We will use an iterative approach to this assignment. Your team will be asked to implement the program over three "sprint" iterations, each about one1 week in length. At the end of each iteration, your team should have a complete, working program. More features and functionality should be added at each iteration. In addition, you are to keep a backlog, a burn-down chart, and hold SCRUM meetings. Each of these will be discussed below.

The following will be the project schedule: (all deadlines are 11:59 p.m. on the given date except the demo, which will be in the Lab final and Lecture final times.)

Date	Milestone
3/26 Thursday	Project 2 Assigned, GitHub repository created with TA added to the project.
3/31 Tuesday	**Initial Design Concept.** Product backlog, initial burn-down chart, and Sprint 1 backlog created. LiveMâché submission due.
4/3 Friday	**Project Proposal** document due. Sprint 1 completion. Sprint 1 Code due.
4/4 Saturday	Sprint 1 retrospective; Sprint 2 backlog created. Recorded Zoom Scrum meeting due. LiveMâché submission due.
4/10 Friday	**User Stories and User Study Proposal** due.
4/17 Friday	**User Study Report** due. Sprint 2 completed. Sprint 2 Code due.
4/18 Saturday	Sprint 2 retrospective; Sprint 3 backlog created. Recorded Zoom Scrum meeting due. LiveMâché submission due.
4/24 Friday	Sprint 3 completion. Code due.
4/27 Monday	**Team retrospective report** due. Individual Report and Peer Review due. Recorded Zoom Scrum meeting due. LiveMâché submission due.
Final	Final Demo During your assigned times (Lab and Lecture)

LiveMâché Platform:
For this project, you will be using LiveMâché, a cloud-based platform providing live, collaborative capabilities for collecting and organizing content, along with writing, sketching, and chat. LiveMâché helps users in discovering and interpreting relationships through visual thinking.

The platform will serve as a one-stop place to keep your team, as well the instructor and TAs on the same page. You will use the platform for collaborative ideation, project documentation, and (likely) presentation. You MUST turn in your mache using the LiveMâché submissions feature at the end of the Initial Design Concept (3/31) and Sprint deadlines (4/4, 4/18, 4/27). We will be evaluating your team's effective use of the LiveMâché platform for collaborative ideation and project documentation.

For more details on how to effectively use the platform, see recommended practices from the Interface Ecology Lab here.
And for the LiveMâché Courses feature, see Student Guide here.

INITIAL DESIGN CONCEPT AND PROJECT PROPOSAL

Although not a lot of work needs to be done by the time of the first deadline (Initial Design Concept), this work will be critical to the success of the project. By the first deadline, your team should have completed the following:

- Appoint one member to be the project manager. Inform the TAs and instructor (Robert Lightfoot) of your choice. The project manager will be responsible for maintaining the backlogs and burn-down charts. This person will also serve as the SCRUM Master during the sprints. As a result, this individual should have slightly less of the programming responsibility when tasks are assigned. **The project manager should be responsible for certain documents owned by the team. Everyone should have access so they can turn in the appropriate documents at the appropriate time.**
- Plan which report will be written by each member. Include this in your initial design concept document.
- Determine project Goal.
- Create a product backlog. Specifically, your team should try to break the overall program into a number of tasks that need to be implemented. You are shooting for a large number of small tasks in this project. The total tasks for a project this size should range in 150-250 individual tasks, not including the reports and demo. You will probably only reach 40-60 tasks for the first deliverable. During the first sprint, you should strive to complete the entire project backlog. Initially tasks might be:
  - Create Dialog box
  Which could turn into
  - Build Drop down menu
  - Build text box with question
  - Build Radio button for yes
  - Build Radio button for no
- Create an initial burn-down chart. This should be an estimate of the overall amount of time (preferably in hours) taken for all of the tasks to complete. Note that burn down chart should be made in *number of hours remaining*, not the number of features, because not all features will take the same amount of time. Your entire team should help make this estimate. One thing to note on hours, if something is designated as taking 10 hours and you have spent 8, verify that you will be done in 2, or update hours remaing!
- Complete the first sprint backlog.

With the Initial Design Concept, your project manager should turn in the product backlog, the sprint one backlog and the initial burn-down chart estimate. You will be turn in your mache, using the LiveMâché submission. Remember to give the instructor and TAs access to embedded Google docs and spreadsheets. The Initial Design Concept document can be an outline of the Project Proposal document due in the next deliverable. This is a group effort and can be in an "Engineering Notebook" style.

Your team should determine the goal for your project and develop an initial prototype. The prototype 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 example, 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.

Note: you can start the User Study earlier than listed in above if you get verbal approval from your TA or the instructor. Also, some of the requirements for the User Study need to be done before the User Study is conducted. (This section is described below in "User Stories and User Study Proposal.") Please review the entire scope of the project to ensure that you are on track to meet the various requirements.

Here is what is required at the Project Proposal deadline:
- **Initial Design Concept:** This is the outline you created earlier.
- **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.
- **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.
- **Planned Report Authors**.
- **Interface Sketch.** You should prepare a sketch/mock-up of your user interface. Hand drawn sketches are highly discouraged unless they are part of the product. If your application will have multiple levels, you may need to have sketches of several screens.

There is a Project Proposal Template that will help guide you in the creation of this document. 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.

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 (4/3). In addition, retrospective, backlog, scrum meeting, and LiveMâché submission will be due (4/4). Turn in your mache using the LiveMâché submissions feature. You can use screen shots or pictures in your document as well.

USER STORIES AND USER STUDY PROPOSAL

The second stage of the project is planning for a User Study and coming up with user stories. The goal of this study should be to evaluate the interface that you developed 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 four stages to a user study: developing user stories, designing the study, performing the study, and analyzing the study. Here we are working on the first two.

Your User Study stage should include the following from each:

- Create at least five user stories of how you envision the product being used.
- 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 be conducted (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. You may also collect quantitative data, although quantitative data is less likely to be useful at this stage. Try to avoid giving "on a scale of 1 to 5" type questions.

This report should include a methodology section:

- 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?

USER STUDY REPORT

You should submit your data in your User Study Report. The data comes from performing the user studies.

(note: we will have to be creative in this area. Contact friends, etc, it is best to use a publicly available URL, virtually and share your screen with them. Teams should also help each other. Your TA can give you the team contact person for user studies. Team one and two can meet on zoom and be each other's users. Then three and four, and so on. Odd and even that is odd + 1. If there is an even number of team, three team can work together.) No one has to meet face to face for user studies, everything can be done virtually.

Note: Instructions for the report.

- Conducting the user study. You are expected to conduct user studies during lab time on the given dates, 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 the class.
- You must conduct studies with at least three users. You are encouraged to do more than three such studies.
- This data and the results will make up the majority of your User Study Report.

- Every person is requested to participate in a user study for at least one 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. You are welcome to go outside the class for additional user studies.

For the User Study Report, you are to include the raw data that you collected from the users. (Raw Data does not count towards the word count described below.) 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 an appendix or 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.

TEAM RETROSPECTIVE REPORT

The same documents can be turned in for each member in PDF format. Each member must turn in the document to get credit. Everything should be PDF files.

At the conclusion of a project, one of the first things to do is complete a retrospective survey. The survey goes out to the full project team. The goal of the survey is to gather team feedback while the project is fresh. Typical questions are focused on project satisfaction, the client and industry, and the work and results:

- Was the project fun and interesting?
- Did it provide opportunities to innovate?
- What went well? What didn't go well?
- What lessons did you learn that you would share with your team?
- What one topic do you want to make sure we address in the retrospective meeting?

We use the survey responses to determine discussion topics for the retrospective report. The raw survey responses are usually only seen by the project manager. In this case share all the data among all team members. Collate the results as an appendix to the document. On a larger scale,

think of what you can take from this project into a semester long capstone project. (This can be a section in the Team Retrospective Report.)

Our retrospective document is the lasting documentation from this process. It's written by the Team and pulls in data from our time and effort spent together, and the team survey responses. Create a title page with credits to the team and the document will consist of a few sections:

- Project overview (A summary of the work and the team)
- Engagement analysis (How was the engagement as a whole? Was the team satisfied? What were the challenges? Did we have the opportunity to innovate?)
- Product analysis (How was our overall final product? What did we learn that we could apply to future work? How could we have done better?)
- Work analysis (Are we proud of what we accomplished? What went well? What could have gone better?)
- Key points to remember for future projects (What did we learn? What would we do differently the next time? What lessons would we share back with our individual teams? The class as a whole?)
- Work effort analysis (How'd we do our time together and time individually? The timeline? What was our breakdown of task per member and how did that differ from our initial estimates?)

This serves as a great way to think critically about a project and also capture a snapshot of the project as a whole. Typically the retrospective documents are shared with the entire company and often refer back to them when starting new projects with similar traits to previous ones. Think of this as a way to take lessons from this class to your capstone or other group projects.

INDIVIDUAL REPORT AND PEER REVIEW

Similar to the final memo of the previous projects, you are to personally reflect on the project. This reflection is meant to be a time to reflect on the processes used in the project. Comparison to the individual and waterfall projects are also welcome. Separate from the reelections you also need to do a peer review of your team mates. I will include a template for the peer review. This is a great place to highlight both the talents and contributions of your team mates.

FINAL PROJECT PRESENTATION

(Please note, this was the original plan. There will likely be changes and we will inform you with plenty of time to make the adjustments.)

Your final presentation will be a Business Casual presentation of your product. You are expected to be able to show the product running in your presentation. You are allowed to have a video of canned portions of your product to show. Videos should not be more than 3 minutes of your allotted time. The last 3 minutes of your presentation should allow for questions from the class.

Your total time is 15 minutes and you should be ready and practiced at using the equipment in our classroom. All final presentations are currently scheduled in the classroom where we have lectures.

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.

Other things to note:
- You will be allowed to **upload a 3 min video presentation** of your project to YouTube. (You get a free account with your ATM account.) You can show your team's video during the final presentation.
- Everyone in your section must be present at the scheduled time for your lab, all students must be present for the lecture portion of the final.
- This presentation will be peer-evaluated, and all students are expected to attend all assigned presentations and participate in the peer grading.

- All team members are expected to participate in the presentation and video if you have one, in some way.

ADDITIONAL PROJECT INFORMATION

Sprint backlogs
Although your sprint backlog has a due date, you are welcome to set your backlog earlier than the deadline, and begin the sprint earlier (though it must be after the completion and retrospective of the previous sprint). Your sprint backlog should be a subset of the product backlog. Note that you may (and probably will) choose *not* to include the entire remaining product backlog in the final sprint if you do not believe it will fit in. You should set your sprint goals realistically. While your project will be graded on overall accomplishment, it will also be graded on how well you met your sprint goals. Setting unrealistic sprint goals may harm your grade more than skipping certain parts of the implementation. Another important idea is to get better at a Sprint after finishing a prior one. Therefore, Sprint#2 should be much better-managed compared to Sprint#1.

Your sprint backlog should be pulled out from your product backlog. From this, you should be able to create an initial burn-down chart.

At the time the sprint backlog is created, you should also set the meeting times for at least 3 SCRUM meeting times during the sprint. Lab times and class times (if class is not meeting) or meeting times before/after class are suggested times for some of these meetings. SCRUM meetings must be held after the backlog is created, which is another reason to set your backlog early. For example, if your backlog is set on Tuesday, you can hold your first SCRUM meeting during the lab on Wednesday. Scheduled SCRUM meetings should not be held on the same day as the backlog is set or the same day as another SCRUM meeting. If you have more than 3 SCRUM meetings, which you are encouraged to have, you may hold them more frequently.

At each of the sprint backlog deadlines, the project manager should turn in the below listed items. Turn in mache using LiveMâché submissions feature (remember to give the instructor and TAs access to embedded Google docs and spreadsheets):

   • The backlog to be used on this sprint
   • A list of which tasks were assigned to which person (at least initially – this could change during the sprint as a result of SCRUM meetings)
   • The initial burn-down chart for the sprint (again, in remaining hours)
   • A list of when the SCRUM meetings for your team will be.
Note that you may set the end of the sprint backlog earlier than the deadline.

During the sprint – SCRUM meetings
Note that the sprints are very short in length. You should have at least 3 SCRUM meetings in each sprint. This usually means meeting at least every other day. SCRUM meetings should be *no more than 15 minutes*. **One meeting each sprint should be a Zoom Meeting, recorded and uploaded to eCampus, all meetings should have written minutes pushed to GitHub**.

Burn-down Charts
The SCRUM master (project manager) should update the *sprint burn-down chart* after each meeting, based on any feedback from team members. We recommend that the SCRUM master keep track of this data in a multi-page google form, where each team member gets a separate page. Each team-member, at the end of each day updates their part in the form (what they think is

remaining for each feature he is responsible for). In the form, another page summarizing the total remaining work for each feature, which makes the production of a graph illustrating progress (or lack thereof) easy.

In addition, the project manager should update the ***product burn-down chart*** at least once during the sprint, and once again at the end of the sprint. Thus, by the end of the project, there should be at least 7 data points on the product burn-down chart: one initial one, one from the start of each of the last two sprints (the start of the first sprint will be the same as the initial one), one from in the middle of each of the three sprints, and one from the end of the final sprint (note that you might not have emptied your backlog).

Sprint Completion
At the end of each iteration (sprint), the source code for the project, to that point, should be turned in. At the end of a sprint, you should have a *completely working* version of the software, although not all features of the product will be included (only those selected for that sprint). This should be turned in by the sprint completion deadline. Note that if the sprint is completed ahead of time, this can be turned in sooner (and progress can be made toward the next sprint.

Retrospective
Following the sprint completion, the team should undertake a retrospective. In the retrospective, the team should get together to determine what went right and wrong during the previous sprint, what changes (if any) need to be made regarding procedure, and what adjustments might need to be made to the product backlog. The project manager should turn in the below listed items for each retrospective. Turn in mache using LiveMâché submissions feature (remember to give the instructor and TAs access to embedded Google docs and spreadsheets):

- An approximately 250 word (1 page double-spaced) write up summarizing the results of the retrospective (what was viewed as good/bad in the process, any changes made).
- An updated product backlog
- The burn down chart from that sprint. It should have at least 5 data points: one initial point, one after each of the 3 SCRUM meetings, and one final one.
- The burn down chart for the project, to this point.

Note that most retrospectives are due at the same time as the new sprint backlog, but **the retrospective should be completed before the next sprint backlog is determined.**

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! You should set up a GitHub repository immediately (at the first deadline).
- Please give the TA and instructors access to GitHub, from the first checkpoint
- The project manager should maintain the backlogs and burn down chart in a location that other team members can easily refer to them. GitHub is an obvious option.
- The project manager/SCRUM master should be given somewhat fewer tasks during the sprint, due to responsibilities for maintaining backlogs and burn down charts. That work will not take a tremendous amount of time, but it is also not insignificant.
- The "customer" in this case is represented by the instructor and TAs. You should involve the customers in your work, getting feedback from the customers often to make sure you are on track and prioritize your backlogs correctly.

Reports:
There are four main reports for this projects.

1. The Project Proposal report,
2. A User Study Proposal and User Stories report,
3. The User Study Results and Design Change report, and
4. The Team Retrospective report.

Each team member should contribute in writing these reports. Reports must be a minimum of 1000 words, not including data or appendices, and turned in for grading. There is a sprint retrospective report due at the end of each of the 3 sprints. There will be individual reports (Including peer reviews) due following the final sprint due date. The details of these reports will be provided in a separate document.

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:

Criteria	Points (%)
Agile computing methodology – setting, using, adjusting backlogs and burn-down charts extracted from retrospect and other forms of documentation. Includes meeting minutes and zoom recordings.	15
Meeting Sprint backlog goals	5
Completeness and quality Mash-up product.	10
Use of 1st , 2nd, 3rd API	10
Project Proposal	7.5
User Stories and User Study design.	7.5
User Study and Changes Implemented.	7.5
Team Retrospective	7.5
Individual Report and Final Demonstration Review	5
Full Usage of GitHub for project.	5
Quality of Project Meeting	5
Quality of Presentation	5
Effective use of LiveMâché for collaborative ideation, project documentation, and (likely) presentation	10
Total	*100*

In addition, the person designated as Project Manager/Scrum Master for the team will receive a separate grade based on the quality and completeness of the reports (backlogs, burn-down charts,

etc.). The Project Manager's grade will be determined by counting the overall project grade with any individual multiplier as 85%, and the project manager portion as 15%.