|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Week** | **Deliverable** | **Requirements** | **Demo** |
| 8/21 | 0 | Semester start | | |
| 8/28 | 1 | Project Kick-off | * Each team member should start to create Slack and GitHub accounts |  |
| 9/4 | 2 | Deliverable 0  Project Setup | * Team has sent mentors list of GitHub usernames via Slack * Team’s copy of template repository set up * Each team member has cloned template project and can run it on local machine using instructions in README (GCP does NOT need to be set up yet) |  |
| 9/11 | 3 | Deliverable 1  ERD | * Entity Relationship Diagram describing your database tables |  |
| 9/18 | 4 | **Milestone 1** | * Have team repository set up * Have pages scaffolded out with routing (can be empty, home page just says “Home”, etc…) * Call example Java endpoint from client (ask the mentors for more details on this in our meeting) | * Demonstrate that all the pages exist by directly hitting the URLs * Display the result of the test endpoint service call |
| 9/25 | 5 | Deliverable 2  Account Creation & Login | * User account creation and login | * Demonstrate creating a user and logging in with that user |
| 10/2 | 6 | Deliverable 3  E2E Service Demo | * At least one full end-to-end (E2E) service is complete | * Demonstrate that an action in the UI will make a request to the backend, requesting data from the MySQL database, and the response will be passed back to the frontend and displayed to the user |
| 10/9 | 7 | **Milestone 2** | * All pages are implemented (sign in/registration, profile page, home page at minimum) * In the service layer, be hitting a MySQL database and be able to create/edit account and add/edit classes/areas of study, add/edit “study partners”, add/edit profile details, create/edit meetups, review tutors, etc. * Have deployed the template project to GCP * Have continuous integration (CI) set up (Git commit triggers a build, which triggers a GCP deployment) * In the frontend, an individual should be able to:   + Register and log-in   + Select user type (student vs. tutor) during registration   + Add/edit classes/areas of study   + Search for and connect with other users   + Create and edit study meetups * Add unit tests in Java layer and add a testing step to the CI deployment pipeline (maintain and update these tests going forward) | * Show the app running live on GCP * Change the message on the home page and commit the change, then check if the continuous integration is working and the message changed on the live site * Demonstrate creating a new account and logging in * Demonstrate creating a new study meetup and providing details (date, time, location, subject, etc.) * Show the deployment pipeline, will all tests passing |
| 10/16 | 8 | *No deliverable* |  |  |
| 10/23 | 9 | Deliverable 4  Recommendations | * Progress with the recommendations system to both find users and meetups * Recommendations should use criteria such as subjects, study methods, type of study partners (students or tutors) | * Demonstrate recommendations for a user based on small set of criteria |
| 10/30 | 10 | Deliverable 5  Notifications | * Progress with the notifications system – viewing messages, marking them as read, deleting * App should notify user if there are unread messages | * Demonstrate notifications for a user when another tries to connect with them or one of their meetups * Demonstrate notification for upcoming meetup |
| 11/6 | 11 | **Milestone 3** | * In the service layer, be able to suggest other users to connect with, schedule, edit, and cancel meetups, send notifications, and submit ratings/reviews for tutors. * Recommendations: A critical component of this project is a sophisticated system for connecting users based on shared interests. Typically, teams achieve this by allowing the individual to enter subjects they wish to study, select whether they are searching for students and or tutors, and set preferences for study methods. Then, the system can recommend a list of other users and the individual can choose to connect with and schedule meetups with any number of them. You can take a different approach from the one above and even add components like a study playlist or location recommendations, but it needs to be able to create a study meetup effectively and also support a large user base. * In the front-end, individuals should be able to:   + Review recommended tutors   + Connect with and join/schedule meetups with recommended tutors   + Get meetup and connection notifications   + Edit or cancel a meetup   + Rate the performance of a tutor after a tutoring session   + Update existing and add new unit tests in Java layer | * Suggestions for students and tutors to connect with related to study preferences and subject * Reminders before meetup * Rating a tutor after a tutoring session |
| 11/13 | 12 | Deliverable 6  UI Hardening | *Note: This is the last deliverable expected for the project until your final presentations* | * As the project nears completion, we want to see how you’ve changed and finalized your UI in preparation for the final presentation |
| 11/20 | 13 | Thanksgiving Break | | |
| 11/27 | 14 | *No deliverable* |  |  |
| 12/5 | 15 | **Milestone 4**  (Due at final presentations) | * Security features (viewing other user’s connections and meetups, only editing user-owned or shared meetups, unauthenticated user viewing the site, etc.)   + Input validation   + User authentication (for example, a validated user can edit their own classes/areas of study but can see others in a read-only format; unauthenticated users see a different view of the site, etc.)   + Password encryption * Final UI/UX design * Bonus features | *The final demo should show off your app completely. However, this is the list of things specifically from Milestone 4 that we would like to see:*   * Demonstrate security * Demonstrate bonus features * The UI should be complete * Product and functionality should be in a complete state |