**Project 1 Summary and Requirements**

**Due Date:** Monday Morning, 5/6

**Teams:** For this project, you can choose to work individually or as part of a pair. Please let me know pairs by **Friday, 4/26.**

**Presentation Style:** Presenters will share their screen and walk through the project’s functionalities via the React-based frontend. 10 minutes max. Everybody should have a speaking role in the presentation, though only one member of the pair needs to share their screen. No slides necessary, though one intro slide is fine. Don’t show code unless asked.

**Executive Summary – Employee Reimbursement System (ERS)**

The goal of project 1 is to create a Java Full Stack Employee Reimbursement System (ERS). The main use case of the ERS centers around Employees submitting Reimbursements that can either be accepted or denied by Managers. The tech stack will consist of a React-Based Front end, communicating via HTTP to a Spring-Based Back end. The database will be either a local or cloud-based Postgres database.

**Employee User Stories**

Users using the application can:

* Create an account (create new User)
* Create a new Reimbursement
* See all reimbursement tickets (only their own)
* See only their pending reimbursement tickets
* [Some other functionality of your choice]
* OPTIONAL: Update the description of a pending reimbursement

**Manager User Stories**

Managers using the application can:

* See all reimbursements
* See all pending reimbursements
* Resolve a reimbursement
  + (update status from PENDING to APPROVED or DENIED)
* See all Users
* Delete a User
  + (should also delete any related reimbursements)
* OPTIONAL: Update an employee’s status to manager

**Validation User Stories (Do Login Last!!)**

Users who are not logged in to the application can ONLY:

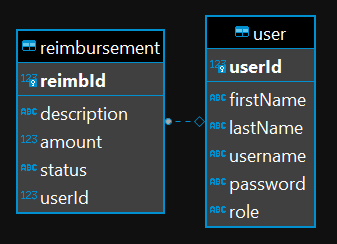
* Attempt to log in.
* Create an account

Users should not be able to access the other user stories before logging in.

**Optional User Stories (Only try these after completing the stories above)**

* Logging of the Service layer with logback.
* Test Suites for the Service layer with JUnit

Both will be required for P2, so this could be good practice!

**Database Architecture**  
  


* Change table columns as you see fit, but keep in mind that these are the diagram shows the absolute minimum requirements.
* Add constraints to the tables as you see fit, and make sure to error handle for them!

Remember to consult the demo, other teams, and google 🎉

and Ben

* How are these graded?
  + If you have all the functionalities working front-end to back-end that’s a solid green
  + If you have almost all of the functionalities (maybe missing 1-2), that’s a lime
  + Missing about 25% of the stories brings you to yellow
  + If you have just a few working, that’s orange territory
  + Very little to nothing done is red