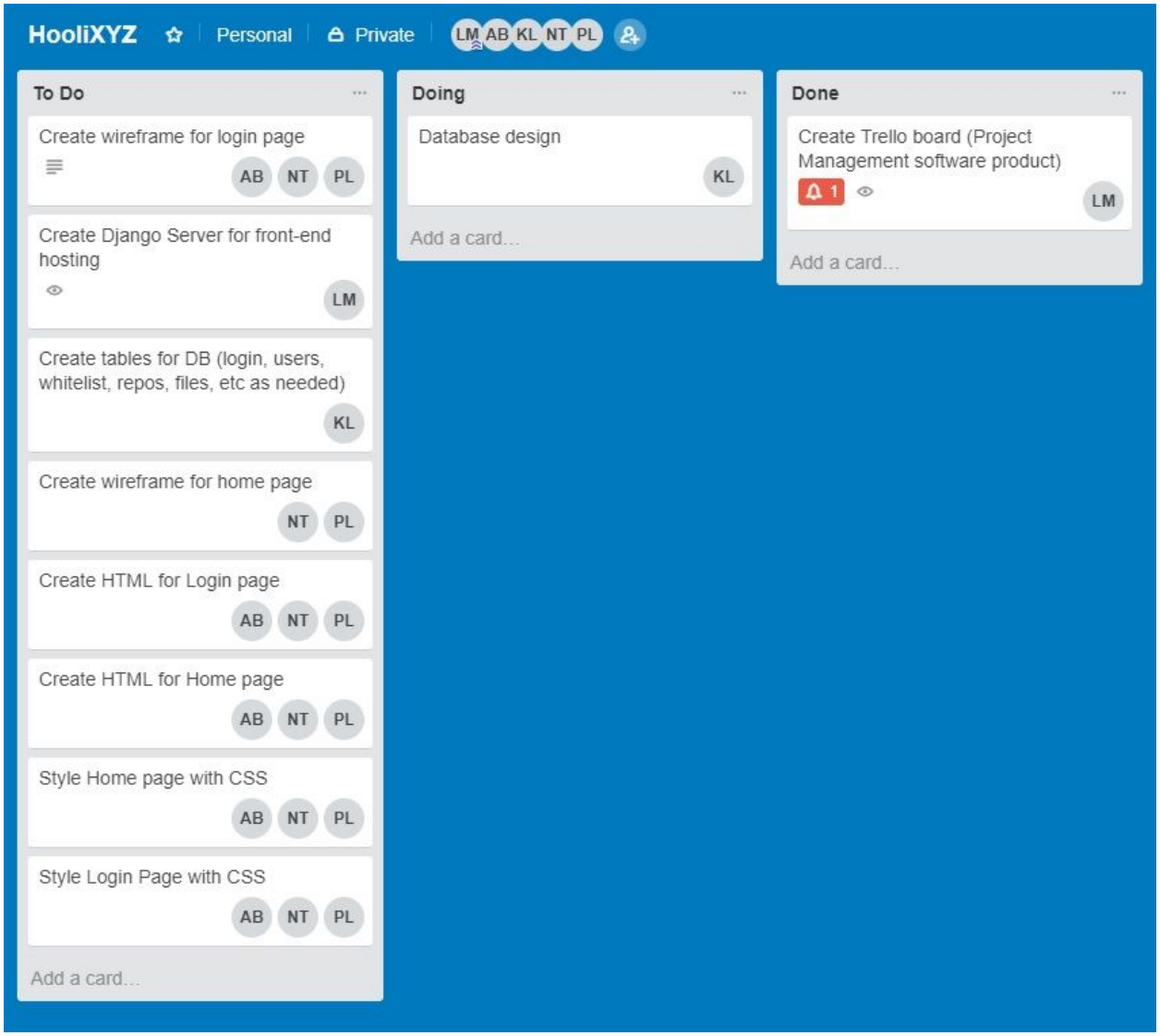
Milestone 2

Team members: Ksenia Lepikhina, Luke Myers, Noel Taterway, Andres Barrera, Peter Lindee
Team: Hooli XYZ

Project Management Tool (screenshot)

Trello is a tool that allows teams to track their progress in a sprint via virtual post-it notes. At the beginning of the sprint, we look at what got finished since the last sprint, add new things to the current sprint, and backlog the unfinished tasks that aren't very important. Our trello looks like the image below:



Product Requirements

The functional requirements specify what a system should be able to do. Throughout Hooli XYZ's sprint discussions, we narrowed down the most important functional requirements. The five most crucial are: to give users to ability to upload and store files to the site, allow users to run files on a copy of the virtual machine on a remote server, allow users to share folders/repositories with other users, allow users to search for other users when sharing, and allow users to sign-up only if they have a colorado.edu email address.

Non-functional requirements are requirements that "define system attributes such as security, reliability, performance, maintainability, scalability, and usability." (https://www.scaledagi leframework.com/nonfunctional-requirements/). The system attributes that Hooli XYZ classified as the most important were: accessibility, capacity (current and forecast), documentation, disaster recovery, privacy, and security.

Accessibility is the quality of being able to be reached. As applicable to our project, this is important for logging in. CU students should be able to login with their CU email and have access to their personal folders/repositories. From there, the user should be able to view their files and share with other CU students if desired. Admin users should be able to login and search for users and view their repositories but not modify them in any way. This can all be accomplished with privilege separation.

Capacity is to what extent we can store the data now and to what extent we can store it in the future. This is related to scalability. As of right now, Hooli XYZ is limited by MySQL and how much it can store. When this product is actually in use, the need to move to a different and more scalable database will be necessary. When we need to scale, we would likely move to a product like Vertica.

Documentation is material that provides a description of the product and how to use it. For our product, documentation is important for teaching a user how to use it. For example, when creating a new account and after initially logging in, the user would be taken to a page that walked through the basics of how to use Hooli XYZ. Documentation is also important for us, the developers, in order to understand how each piece of our product works together as well as how to run it.

Disaster recovery is crucial if Hooli XYZ was to be made into a usable product. This requirement looks at what would happen if data was lost (by an external or internal source). Our prevention/mitigation strategy would be to backup the data regularly to minimize lasting damage. This is highly important for the liability of the company.

Privacy ties back to accessibility. The key piece of the privacy requirement is, once again, privilege separation, as well as the ability to create private repositories. For Hooli XYZ, private is the default.

Security is how well we protect any data that we collect. Our product should mitigate SQL injection and cross-site scripting. This is easily done with prepared statements and user input sanitation. How well we protect our data is also valuable for liability as well as maintaining the integrity of our data.

These six requirements are significantly important for the success of Hooli XYZ in the long run. These requirements help us have a clear goal for which features need to be

implemented. The features are discussed in more detail in our three week plan on the following page.

3 Week Plan

Hooli XYZ plans on having a sprint meeting once a week on Mondays at 6pm for the remainder of the project. Peter, Noel, and Andres are working on the front end. Luke is working on the back end. Ksenia is working on the database. The tasks for each sprint are laid out below.

I. Week 1
   A. Front End Team
      1. Decide on template for users to interact with (due: 06/21)
      2. Finish the template chosen (due: 06/25)
         a) Create general page for files and folders (due: 06/22)
         b) Include a header for users name (due: 06/23)
         c) Include a search bar (due: 06/24)
      3. Create a login template with username(and/or email) and password with enter button (due: 06/25)
   B. Back End Team
      1. Create Django Server capable of hosting pages (due: 06/23)
      2. Create image of VM terminal and link to server (due: 06/25)
   C. Database Team
      1. DB design
         a) Configure MySQL (due: 06/22)
         b) Consider tables that need to be created (due: 06/25)

II. Week 2
   A. Front End Team
      1. Make majority of page functional (due: 06/30)
         a) Login (due: 06/27)
         b) Buttons (due: 06/29)
      2. Integrate with back end (due: 07/01)
      3. If there are no problems integrating with back end, start to make page cleaner and smoother. (due: 07/01)
   B. Back End Team
      1. Integrate server with front end (due: 07/01)
   C. Database Team
      1. Create DB
         a) Implement the tables (due: 07/01)

III. Week 3
   A. Front End Team
      1. Testing functionality, completing integration with back end (due: 07/03)
   B. Back End Team
      1. Finalize front end integration (due: 07/03)
      2. Integrate back end with database (due: 07/03)
   C. Database Team
      1. Integration with back end and front end (due: 07/03)

Stand-Up Meeting for Sprint 1

- Luke Myers (Scrum master)
    1. What have you completed? Nothing.
    2. What will you complete before the next meeting? Creating the server and running a VM on the server.
    3. What roadblocks will you face? Getting the VM to run on the server.

- Peter Lindee
    1. What have you completed? Nothing.
    2. What will you complete before the next meeting? The wireframe and HTML for the login page.
    3. What roadblocks will you face? Deciding on a template to base the HTML on.

- Andres Barrera
    1. What have you completed? Nothing.
    2. What will you complete before the next meeting? The wireframe and HTML for the home page.
    3. What roadblocks will you face? Creating the search bar.
- Noel Taterway
    1. What have you completed? Nothing.
    2. What will you complete before the next meeting? The CSS styling for the login and home page.
    3. What roadblocks will you face? Creating the basic template for the web application.
- Ksenia Lepikhina (Product owner)
    1. What have you completed? Nothing.
    2. What will you complete before the next meeting? The database design for the web application.
    3. What roadblocks will you face? Determining which tables need to be created and their cardinality.

Summary of stand up for sprint 1:

As of right now, we have not made great progress on the project. This sprint was productive because we laid out a plan for which features must be built by the next sprint. What didn't go well was that we didn't start developing Hooli XYZ prior to the meeting and therefore were limited in our discussion for what we have completed. Our team could improve on discussing one piece at a time. Instead of discussing the front end and back end simultaneously, we could discuss each piece individually and then integrate the three sides (front end, back end, DB) at the end of our discussion.

Agile Sizing - End of Lab #1

Team member names: Ksenia Lepikhina, Luke Myers, Noel Taterway, Andres Barrera, Peter Lindee

Story 1:
"As a CU student, I want a website where I can store my files."

- Create personal pages for each user - 2
    - HTML, CSS, Bootstrap, Javascript, MySQL, Python
- The user can login and view their folders - 8
    - Python, HTML
- By clicking the folder, they enter a new page where they can see all their files - 1
    - Python, HTML, Javascript

Story 2:
"As a CU student, I want a website where I can share certain files and/or files with other students."

- Create private pages for each user - 2
    - Python, HTML, Javascript
- The files should have a share button that, when clicked, opens a pop up that allows users to enter an email of a user they want to share it with - 8
    - Python, HTML, Javascript
- After it's shared with another user, the user should have the folder/ "repo" appear on their page - 8
    - Python
- Copy a folder from one person's page to another - 5
    - MySQL

Story 3:
"As an admin of Hooli XYZ, I want to be able to see all of the folders that a user has by entering a username in a search bar and going to their repo."

- Enable permissions - 3
    - Python, MySQL
- As an admin there should be a search button where they can find users by email or username and access all of their folders and files (READ ONLY) - 10
    - Python, HTML

Story 4:
"As a CU student, I don't want to see anyone else's repos unless I have access to them."

- Don't allow a user to search for other users and view their repos - 3
    - Python
- The only thing a user should see when they login should be their folders, files and a share button where they can enter an email to add a user to a repo - 2
    - Python, HTML, Javascript

Story 5:

“As a CU student, I want my files and folders to be secure.”

- Create a login page - 2
    - HTML, Javascript
- Enable permissions - 3
    - Python, MySQL
- Allow users to edit files if the owner shared them with them - 8
    - Python, HTML
- Allow admin to read all files but not modify them - 10
    - Python, HTML