Project Milestone 3
Justin Ridenour, Josh Davis, Ken Johnson, Torrin Conrath

High Level Design:

When constructing the high-level design for the canvas kanban board multiple applications need to work together to create a fluid process. How this kanban works is that it will be an online application where the client will have to input their canvas API key (.ics). This then would lead to a fetching sequence where the API key would export the client's data in a database. This hud which has the necessary blocks to separate the appropriate data from the information is viewable. Then the database must have the ability to sort the data depending on what the user wants.

The best design architecture to fulfill the project specifications would be a **Model-View-Controller architecture.**

Firstly, the user should be able to both view and control the data on their Canvas Kanban Board. When the user is in the interface, they would need to select how they want to sort their assignments. This would require a controller to change the model of data. Then this model would update the view of the data that the user sees.

- There would be two models, both the Canvas API Model and the Kanban Board Model.
- There would be one view, with the Kanban Board View which displays the assignments and their priority.
- There would be two controllers, one handling the input of the API key and the other allowing the user to switch the sorting method for the assignments.

The Model-View-Controller architecture is the most common architecture for web applications which our application is set to be. This architecture allows seamless interactions between the database and the user with classes. Separating these functions into two different MVC architectures (API fetching and sorting assignments) will allow the application to be both adaptable and clear in its functions. The Model-View-Controller architecture allows the Canvas Kanban board application to be the most structured and scalable with its development and seamless with the interactions between the user and data.

Low Level Design:

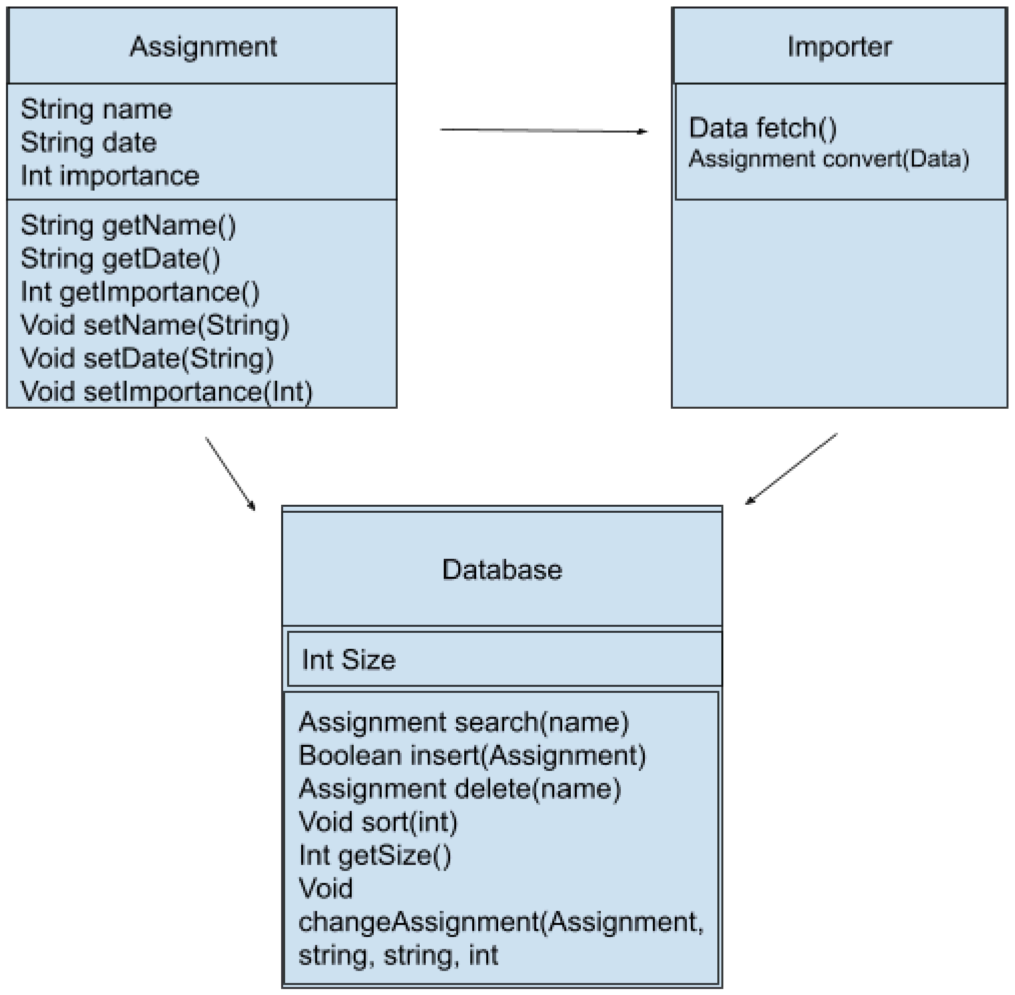
The design pattern that will be most effective for completing this project is the structural family. We are designing our kanban board to automatically import assignment data from canvas, therefore we will need to grab data directly from the canvas api. One of the most important tasks for this project will be to convert data collected from the canvas api into usability data for our kanban board data structure. The adapter aspect of the structural family will be helpful for this implementation. Our project will have two extremely different interfaces (in the API fetcher and the kanban board data structure) and effectively making these 2 interfaces compatible will be the central focus of our design.

Design Pseudocode:

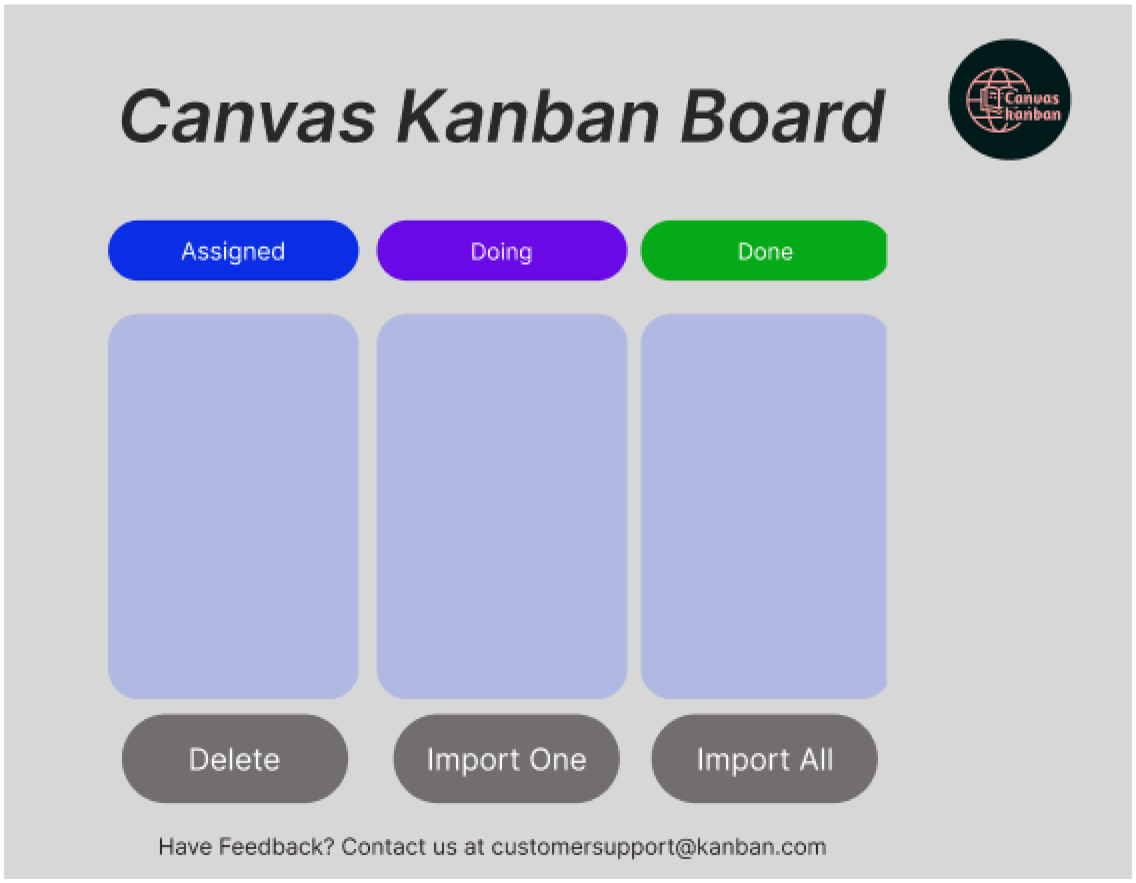
```
Import method
For each assignment in canvas
    data a = importer.fetch()
    If (not currently in database)
        {
        Assignment assignment = importer.convert(a)
        Database.insert(assignment) //inserts the converted assignment into the
        database
        }
Sort the database by preferred sorting method
```

Class Design:

Below is a preliminary class design for our project.


Assignment
String name
String date
Int importance
String getName()
String getDate()
Int getImportance()
Void setName(String)
Void setDate(String)
Void setImportance(Int)
Importer
Data fetch()
Assignment convert(Data)
Database
Int Size
Assignment search(name)
Boolean insert(Assignment)
Assignment delete(name)
Void sort(int)
Int getSize()
Void changeAssignment(Assignment, string, string, int

Design Sketch



In this mockup of the user interface we have a few distinct features that are important to the functionality of the app. There are three columns that are typical of the Kanban style method which are assigned, in progress, or completed. Users will be able to drag and drop between the columns to accurately show the completion of their tasks. The design is minimalistic and intuitive because an approach of “less is more” allows the user to get the hang of the app a lot quicker. We also added a feedback email in case there were any bugs the user experienced which allows for the constant improvement and iterative design of our application.

Process Deliverables:

Kanban Method (list of remaining tasks ranked by priority):

1. Integrate Canvas API
2. Implement remove assignment feature
3. Fix assignment duplication bug
4. Implement drag and drop between columns
5. Implement new sorting methods(e.g. Sort by class importance as determined by the user)
6. Improve status suggestions