The A-Team

UMBC Textbook Marketplace

Tuc Nguyen - Amra Mo - Jun Kwon - Hao Wu

Amir Beshir - Sophia Shin

IS 436 - Structured Systems Analysis and Design

Deliverable V - “Closing: User Interface Design, Program design and System Implementation” (D5)

May 9, 2019

**Table of Contents**

User Interface Design…………………………………...……………...….………..….3

Interface Standards

Program Design…………………………………………………...………………....…5

System Implementation…………………………………………………...………...….7

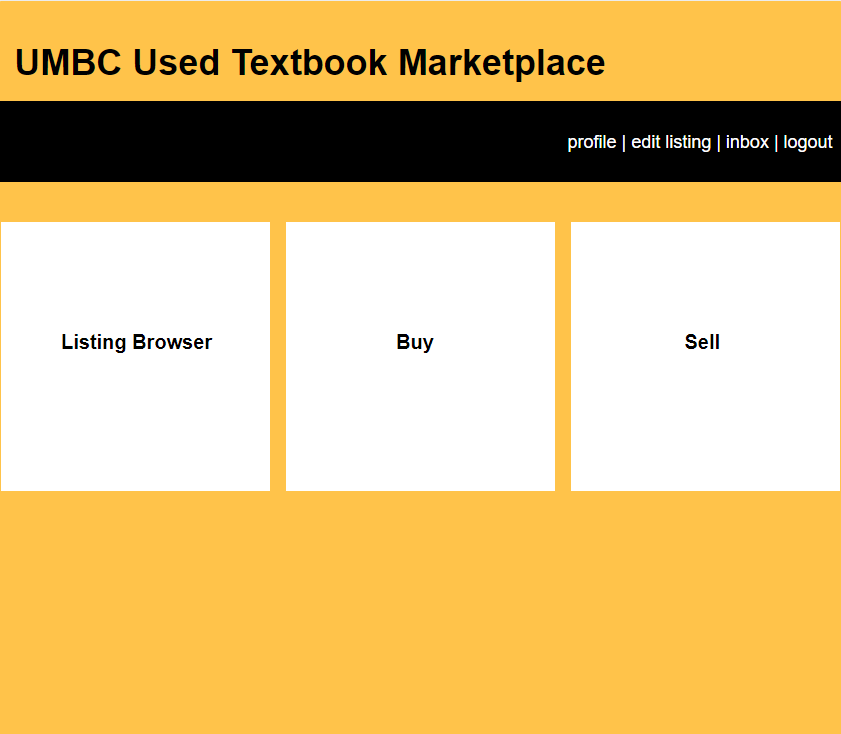
Project Work Plan…………………………………………………...……………….....8

Kanban Board…………………………………………………...………………....…...8

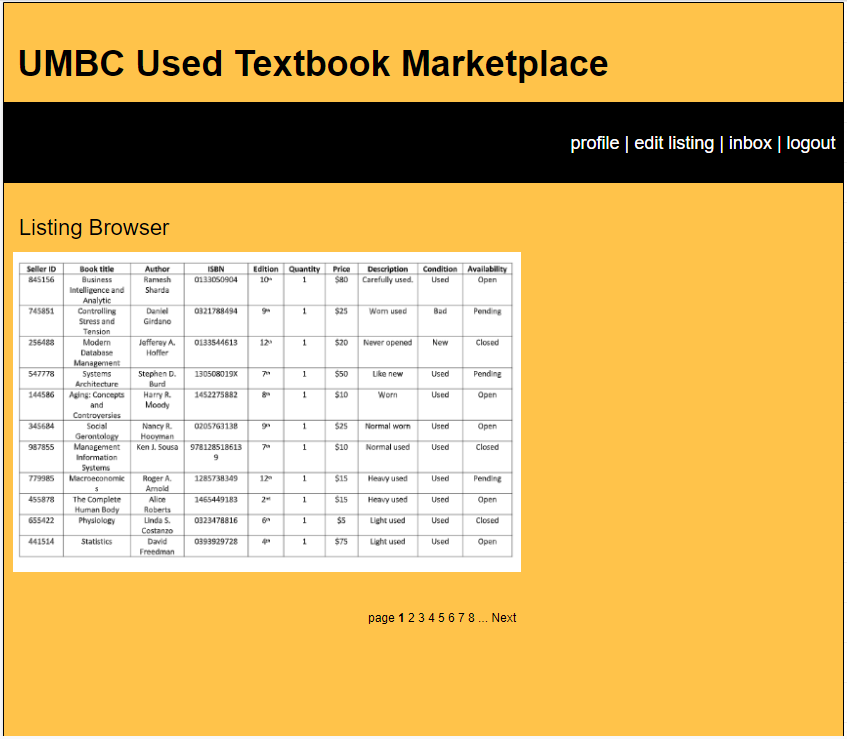
Discussion…………………………………………………...………………....……….9

**User Interface Design:**

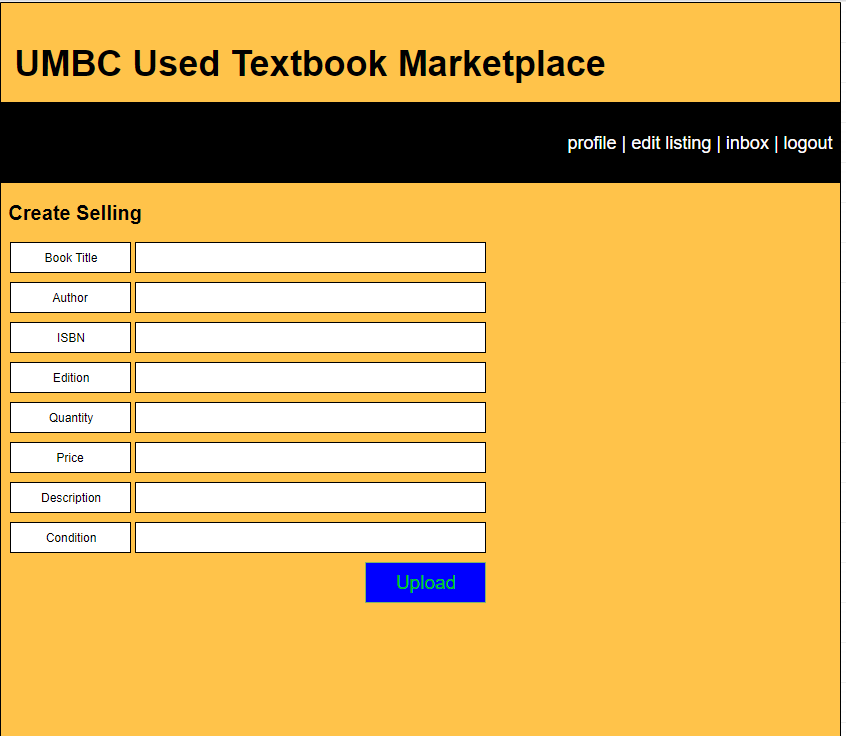
Interface Screen Prototype:



Listing Browser Prototype:



Sell Listing Prototype:

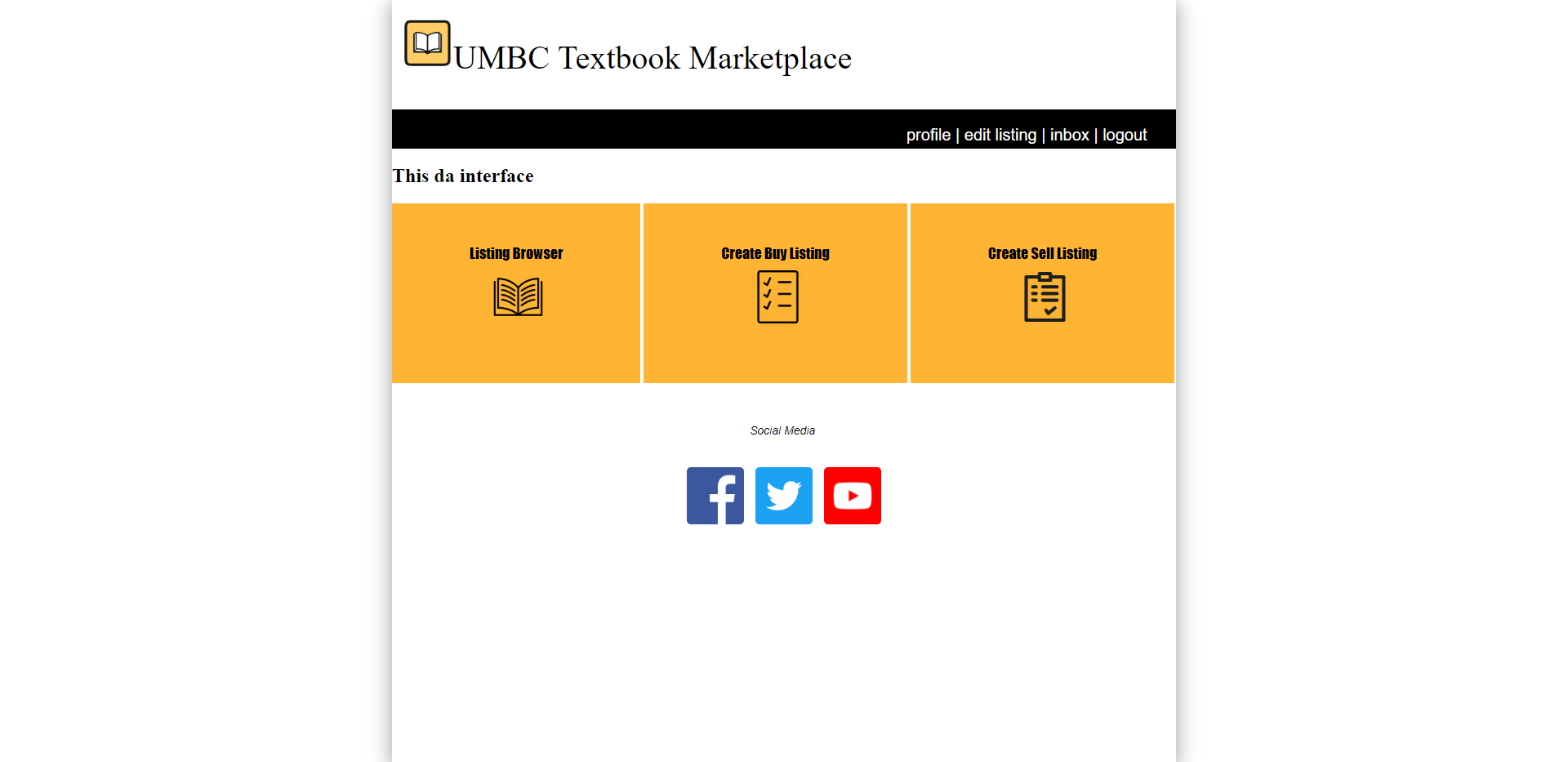


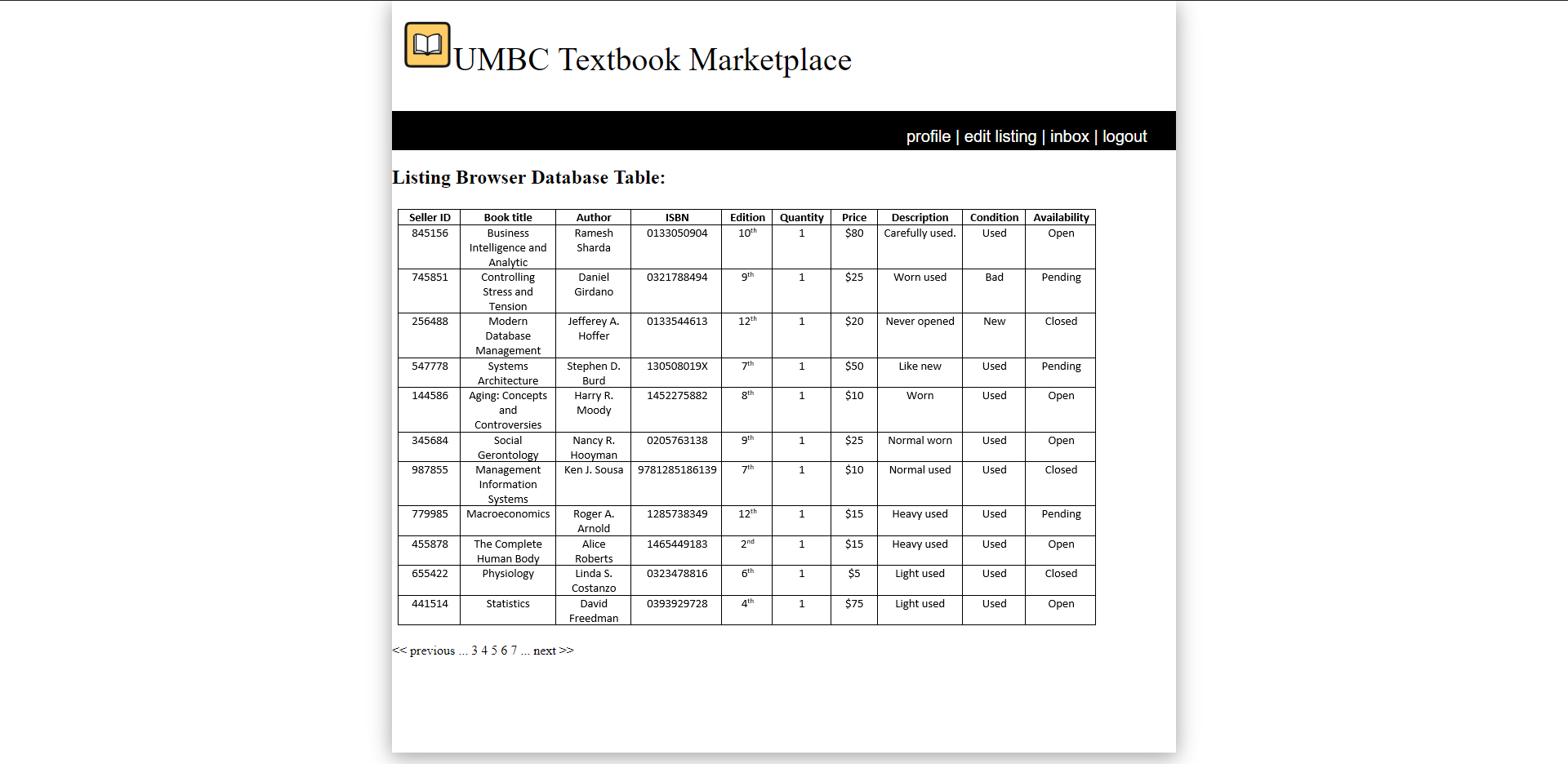
**Interface Standards:**

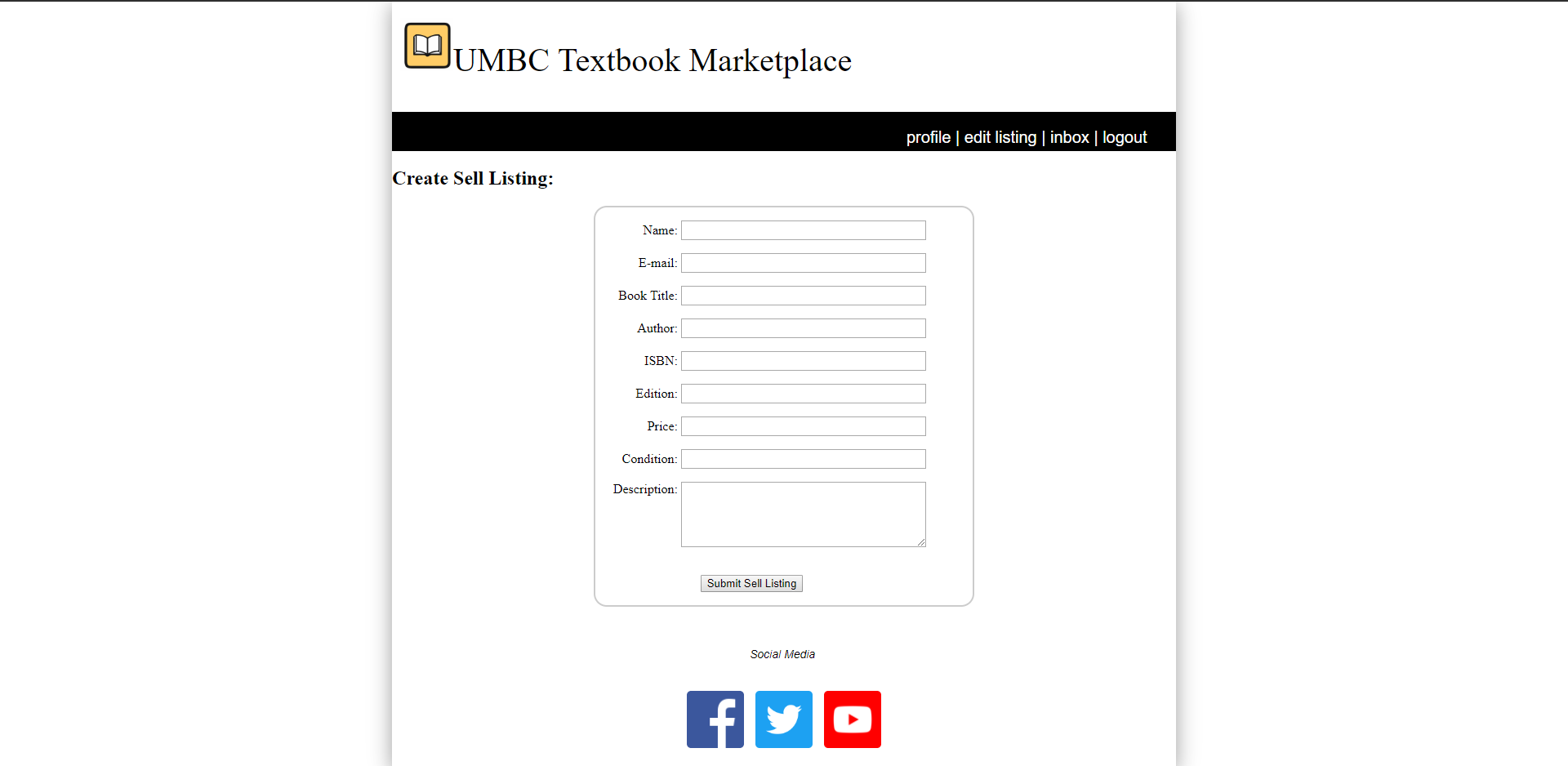
The interface should allow users to choose a list of options they want to complete (Listing Browser, Create Sell Listing, Create Buy Listing) in the form of a button. In addition, the user should be allowed to see a compiled list of books available or pending. All information is stored within the database. Create Sell Listing page will provide a fillable form for users to input their information to upload their book information to the database in order to be viewable by users on the Listing Browser Page. Each HTML page should not be complex and be relatively easy to navigate.

**Program Design:**

The program was developed on HTML, a link will be prepending on GitHub for a live view of the working website.

****

****

****

**System Implementation:**

The system requirements needed for our Textbook Marketplace is twofold. The software on the front end and the back end, and the hardware needed to support such software. However, there are two often overlooked parts of system implementation: quality assurance and maintenance. Quality assurance is important because if there were bugs or vulnerabilities in the system, our users’ sensitive data such as passwords or contact information would be liable to attack. Maintenance is hugely important as well, as the costs associated with starting a system only occurs once, when the system is set up -- the rest of the costs until the end of time can be considered maintenance. Our system requirements are not huge by any stretch of the meaning, but regardless they are critical to meet for operation.

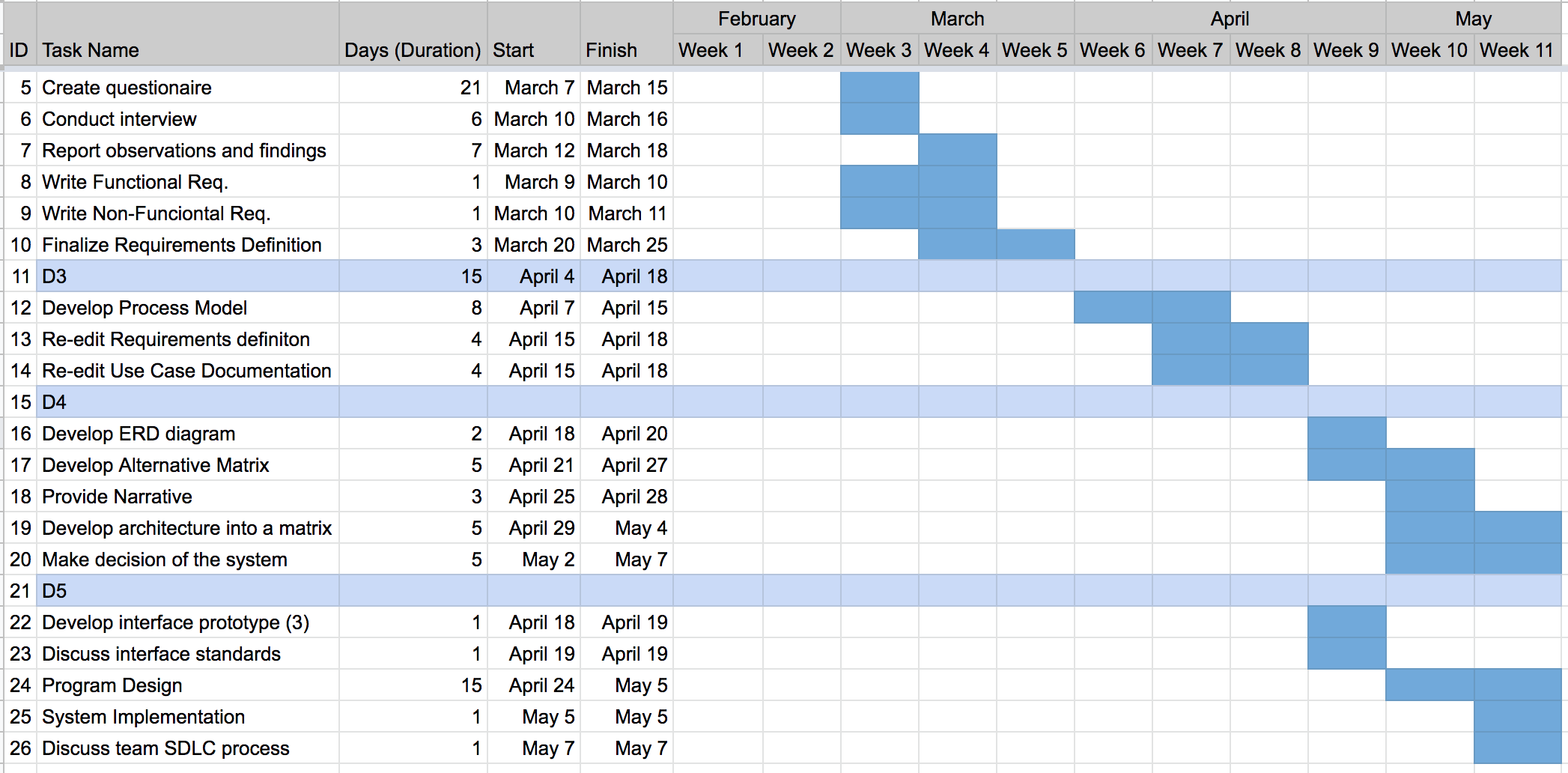
Our software and hardware requirements are simple. As a web application, we do not need a large server to host our website and database, nor any specialized software. In fact, a simple mid-range desktop connected to the internet can certainly meet our requirements. However, a local desktop is liable to power failure, physical security vulnerabilities, or simply just neglect. There are plenty of cloud services available to host a virtual machine for us at a low monthly cost, with 24/7 uptime and maintenance and support, a much more viable option. As for software, Windows, having licensing fees, is not required; Linux is free and can do all of the tasks that is needed in terms of web hosting and database management. However, there is the tradeoff of usability -- training will most certainly be needed for Linux, which can potentially cost more than Windows licensing. In this case, the Windows licensing makes the most sense, as it is a known cost and an operating system our programmers are comfortable with..

Finally, quality assurance and maintenance. Quality assurance will be done by our team in order to check for security vulnerabilities or bugs in the coding. Maintenance is a non-issue with the cloud computing services, as most of the costs are rolled into the cloud services fees. However, it is possible that future expansion, such as coding a phone application to integrate with our system, can rise the amount of resources we use, thus increasing our maintenance fees. For now however, maintenance is simply a bill.

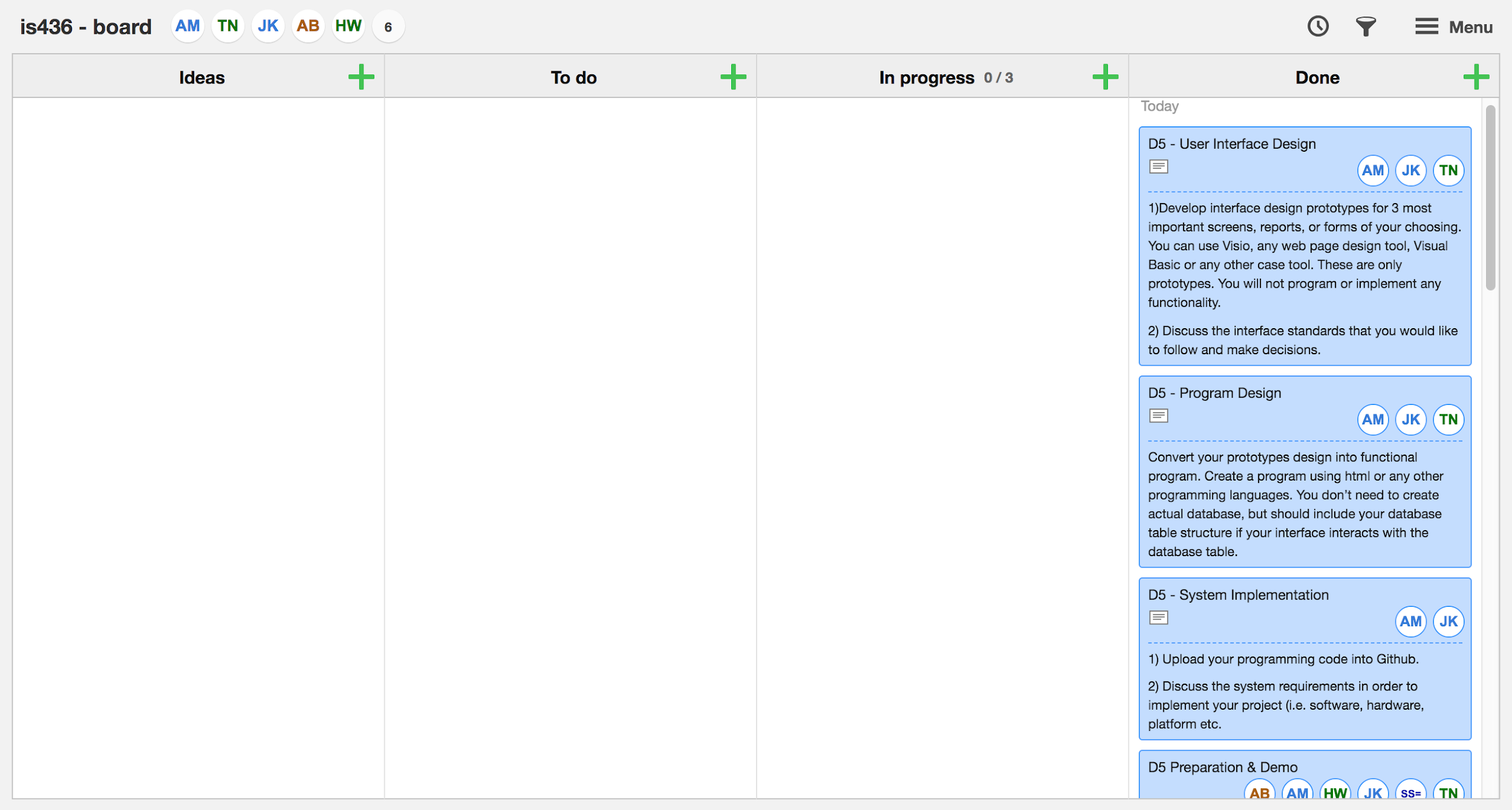
<https://github.com/jkwon41/IS436>

<http://htmlpreview.github.io/?https://github.com/jkwon41/D5_IS436/blob/master/interface.html>

**Project Work Plan:**



**Kanban Board:**

****

**Discussion - Team Experiences through the SDLC process:**

Our experiences through the SDLC process ensured us as a team to implement the systems and applications in an adequate way. We were able to clearly outline the process for our system and was able to define and enforce each goal to incorporate our key features prior to moving on to the next phase of the project.

We found it rewarding to take each step of the process to build a proper system. We first prepared a system request analysis for the project to instantiate the purpose of developing the application. Upon proposing our request, we were able to build upon the project in terms of creating our requirements definition and gathering field information through interviews to gather those requirements. We then produced use case documents to cover several functions in the functional requirements. From there, we were able to develop our process models using the diagrams, and through our findings, we were able to adjust the requirements definition and use cases to fit our system more accordingly. Next, we developed our data models and an alternative matrix to come together and agree on a finalized team matrix. We then designed the architecture of the system by developing it into a matrix, where we were able to list non-functional requirements of our project under the four main categories. We came to a decision of the system architecture and explained our justification for it. For the next phase, we first drew a prototype for our interface design. When we came to the final decision, we could move on to implementing the actual program. We used HTML and CSS to design the website the application would be hosted on. We tested the program on several occasions to ensure that it would work on most browsers. Finally, we were able to launch a working demo of the project.

There were challenges we faced as a team that came along. The issue we faced most, was key timeframes for completion. It was crucial to complete each objective, yet there were delays in terms of revisiting prior requirements and planning along the way. Our other issue was the planning of events and figuring out the capacity of how far we could take our project but with the least optimal way. We observed tradeoffs and the reasoning behind it when completing a task.