**Academic Pathway Optimization**

Goodfellas

IS436 - 03

Deliverable 4

May 1st, 2019

**Project Members**

Paul Lyon

Tanner Yatsko

Jas Singh

Zach Jaquet

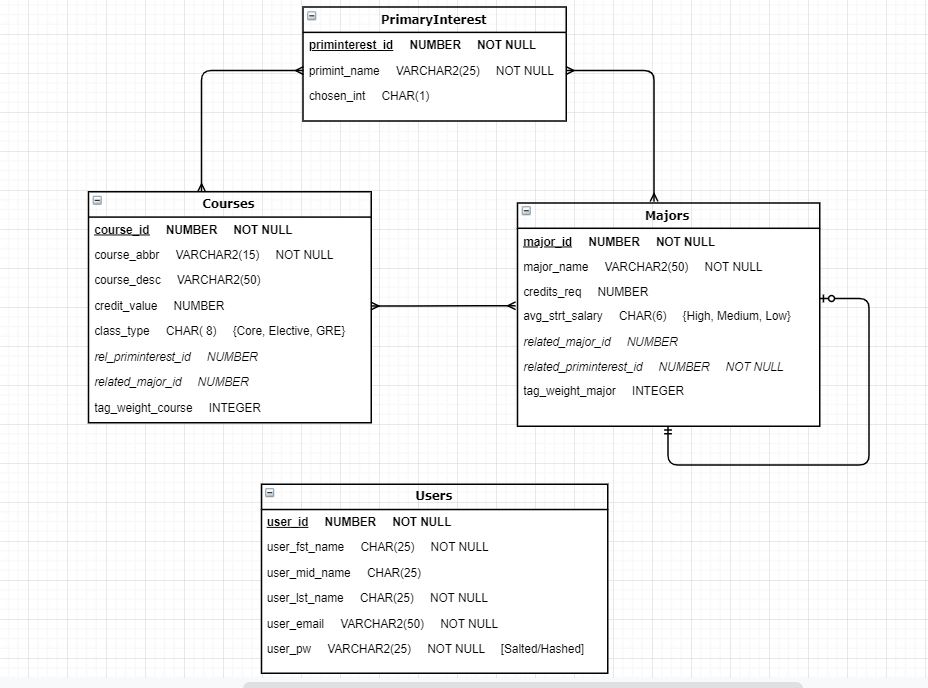
Benjamin Wang

Densu Kurian

William Chanmugam

Niels Verhoeven

**Entity Relationship Diagram**

****

**Entities and Relationships**

**Tables:**

Users: This table shows the users registered within the database. This consists mostly of administrative users but is open to user information that is not strictly administrative. This table has no relationship to any other tables in our RDB.

* Primary Key: User\_ID

PrimaryInterest: This tables houses the options of interests that a student/user may select. These options tie into majors/courses that will be recommended to them.

* Primary Key: PrimInterest\_ID

Courses: This table displays all courses that are associated with listed majors a student/user can select from or would need to complete major requirements.

* Primary Key: Course\_ID
* Foreign Key:
  + Rel\_PrimInterest\_ID (to PrimaryInterest table)
  + Related\_Major\_ID (to Majors table)

Majors: This table shows all available majors at UMBC, that ties in with interests a student/user selects.

* Primary Key: Major\_ID
* Foreign Key:
  + Related\_Major\_ID (to Majors table)
  + Rel\_PrimInterest\_ID (to PrimaryInterest table)

**Added Constraints:**

* Courses:
  + Class\_Type field can only be ‘Core’, ‘Elective’, or ‘GRE’ in value
* Majors:
  + Avg\_Strt\_Salary field can only be ‘High’, ‘Medium’, or ‘Low’ in value
* Users:
  + User\_PW field must be hashed or salted to prevent potential for all passwords being jeopardized by hack.

**Relationships:**

* PrimaryInterest to Courses (Many-to-Many) - This is a many-to-many as a single course may have multiple associated interests, and a single interest may have multiple associated courses.
* PrimaryInterest to Majors (Many-to-Many) - This is a many-to-many as a single interest may be associated with multiple majors, and a single major can be associated with multiple interests.
* Courses to Majors (Many-to-Many) - This is a many-to-many as a single major can have multiple associated courses, and a single course may be associated with multiple majors.
* Majors (self-relationship) - The Majors table has a relationship with itself as a single major may have a major that is similar (this is not mandatory).

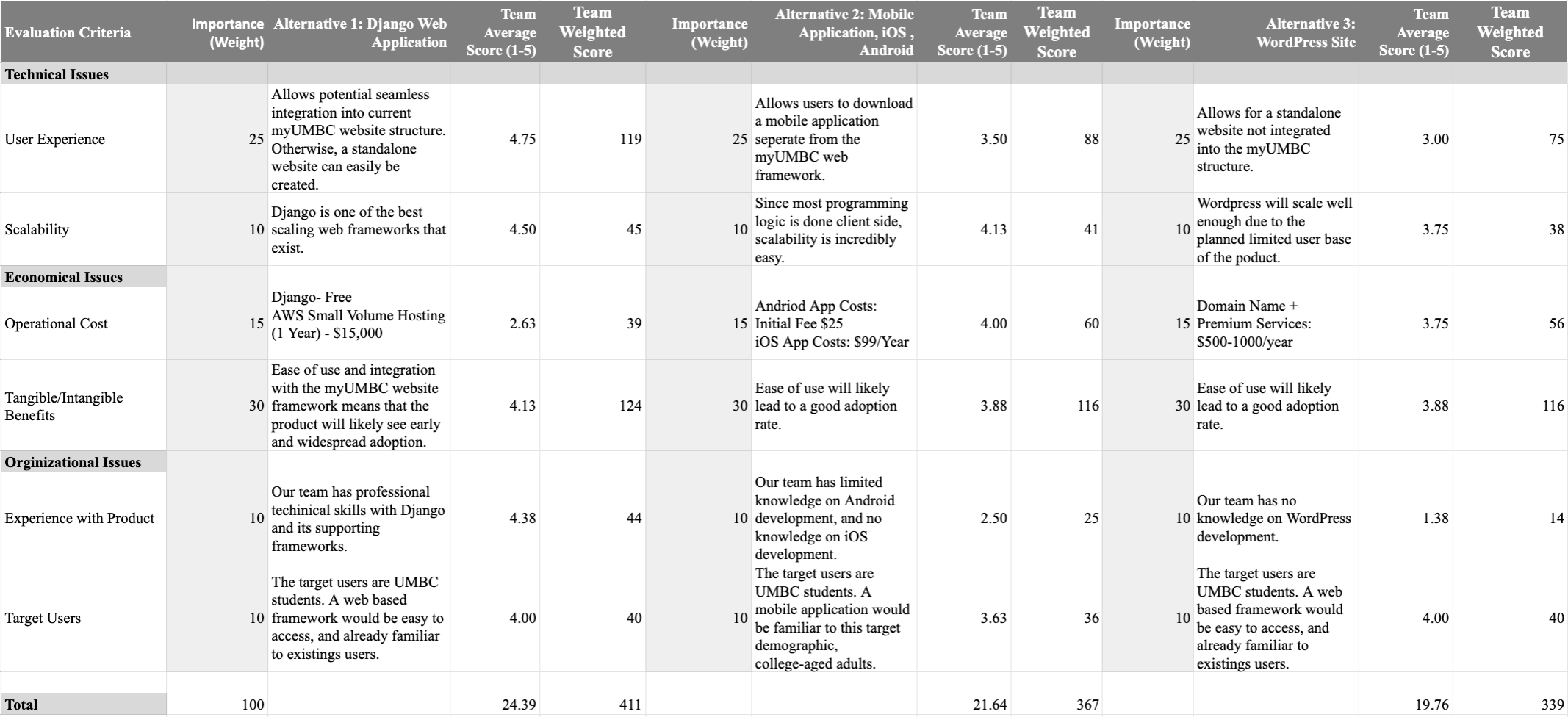
**Alternative Matrix**

****

**Alternatives:**

For our alternative matrix, we decided on going with a Django Web Application, a mobile (iOS, Android) application, or a WordPress site. A Django Web Application is based on an open source and python-based web framework. Mobile applications, which would primarily be run on Android and iOS, would be used mainly on phones and tablets. They would be developed and sold on the Google Play and Apple App Store. Smartphones are widely adopted across the age of users we are targeting, giving it an edge in terms of adoption. Finally, we looked at WordPress, which is a relatively easy method of creating a site using templates. It would be easy to use for users, but functionality is limited creating scalability and specialization issues. Please view appendix page x-x for individual alternative matrices filled out by team members. The team matrix, a consensus between all team members is on the following page.

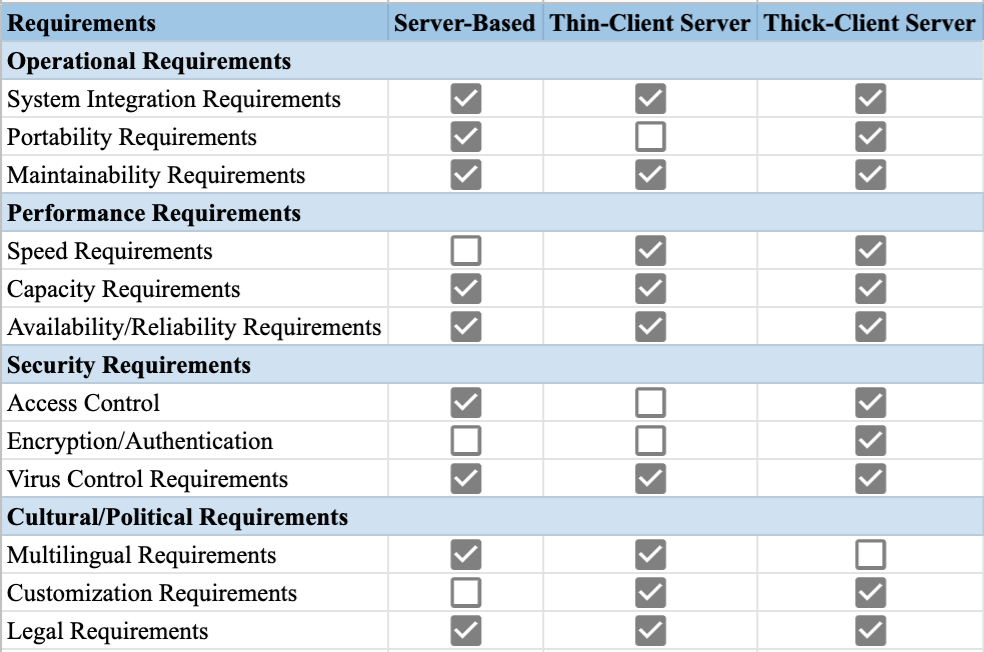
**Team Alternative Matrix**

****

**Alternative Decision:**

For our Team Alternative Matrix, we averaged what all 8 team members put down for their individual matrices. We had three alternatives; Django Web Application, Mobile Application (IOS/Android), and a WordPress Site. After the calculations we concluded that the most ideal alternative would to build a Django Web Application. One reason is because most of the team involved with the implementation and programming of the site has background knowledge and/or experience working with Django. User experience is a big role in the success of our project we rated Django as the best to build a fluent and clear UX. Also, the scalability capacity of Django is ideal. On the other hand, the operational costs of Django surpassed the others, but we felt that price was not necessarily accurate and could be up for debate if implemented differently.

**Architectural Matrix**



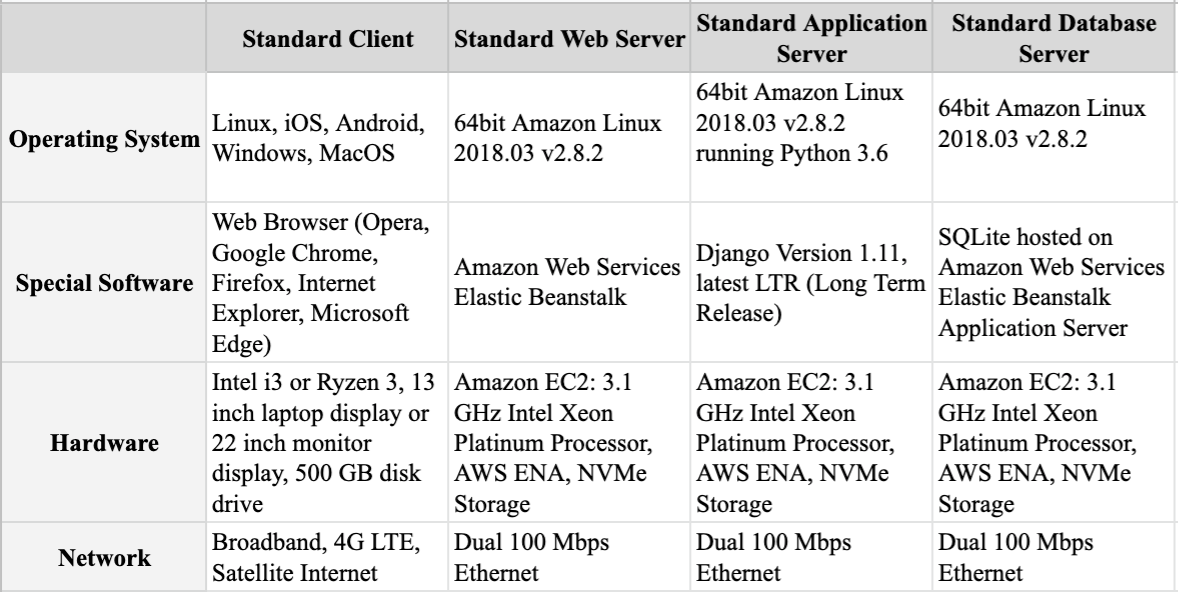
**Architectural Fits:**

All three meet the system integration requirements because depending on the design a system can be made to integrated into any of the architectures.Thick and client server meets the portability requirement as we would have a solution with each device and/or OS.Maintainability is obtainable amongst all three servers from small to large scale changes. Thin and thick client servers easily meet speed requirements (~2 seconds) for site functions. The volume of users is relatively low; therefore, any architecture would probably work.For availability and reliability, all three would fit, as slight downtime would not cause significant architectural issues to the core of our system.Server data is stored behind administrator interface that client should not have access to, therefore server-based or thick-client architecture would work best for access control.Nothing needs to be encrypted or authenticated from user end, so thick-client architecture would work fine.For virus control, the data is not sensitive because no personal information exists, therefore all three would work. Language is an easy functionality to change from client side if given the option so that would be most ideal. Customization is rather accessible to the client and can customize as wanted, so thick or thin would work best. Finally, all the architectures are legal, therefore meeting legal requirements.

**Decision:**

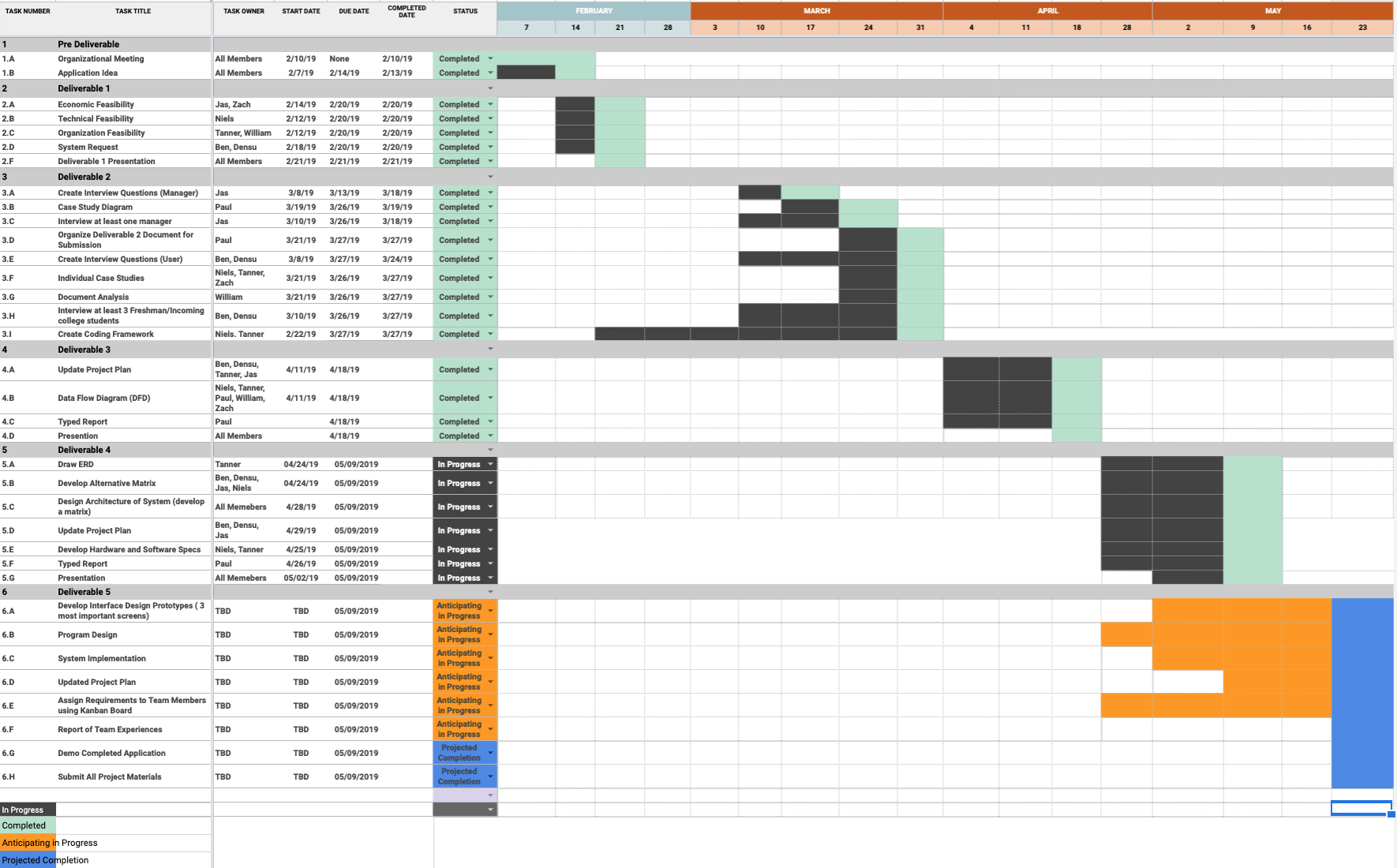
After weighing the different architectures and how they met our needs, we decided to go with a thick-client server architecture because it met the most amount of our operational, performance, security, and cultural/political needs. We found it most important that the site be easy for the user to use, but at the same time secure enough that the users can’t mess with the tables and files that the site uses. These requirements are met with a thick-client server architecture. Speed and capacity can be easily met with modern technology at a very low cost with such a small site. Though some cultural and political requirements would be a little more difficult to implement such as multilingual functionality, the gap is not huge and could still be easily done.

**Software and Hardware Specifications**



**Note:** Our server side will be run by Amazon Web Services which use EC2 instances. Amazon keeps most of the specific hardware components as proprietary information, therefore the capacity and various other specs are unknown for a portion of the hardware specifications.

**Updated Project Plan**

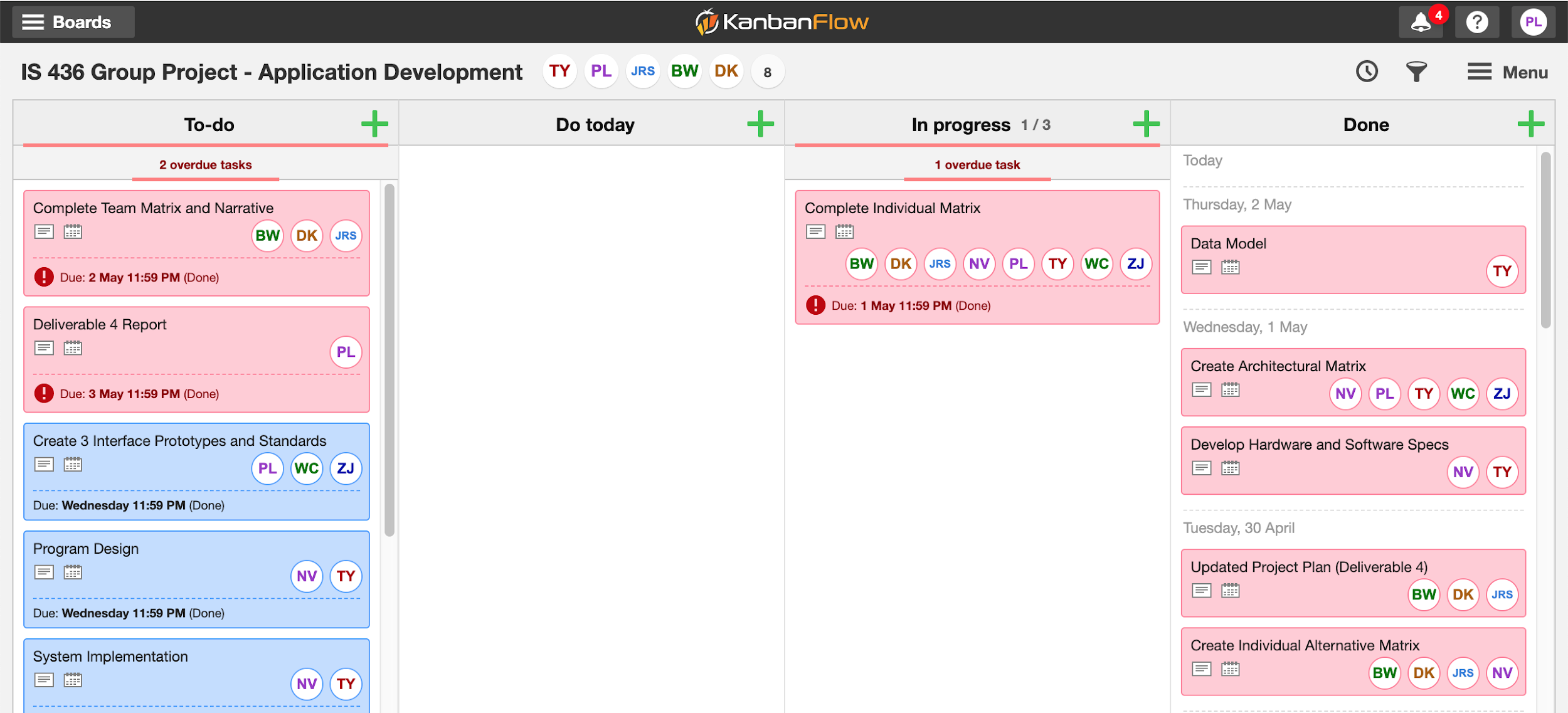


**Gantt Chart Summary:**

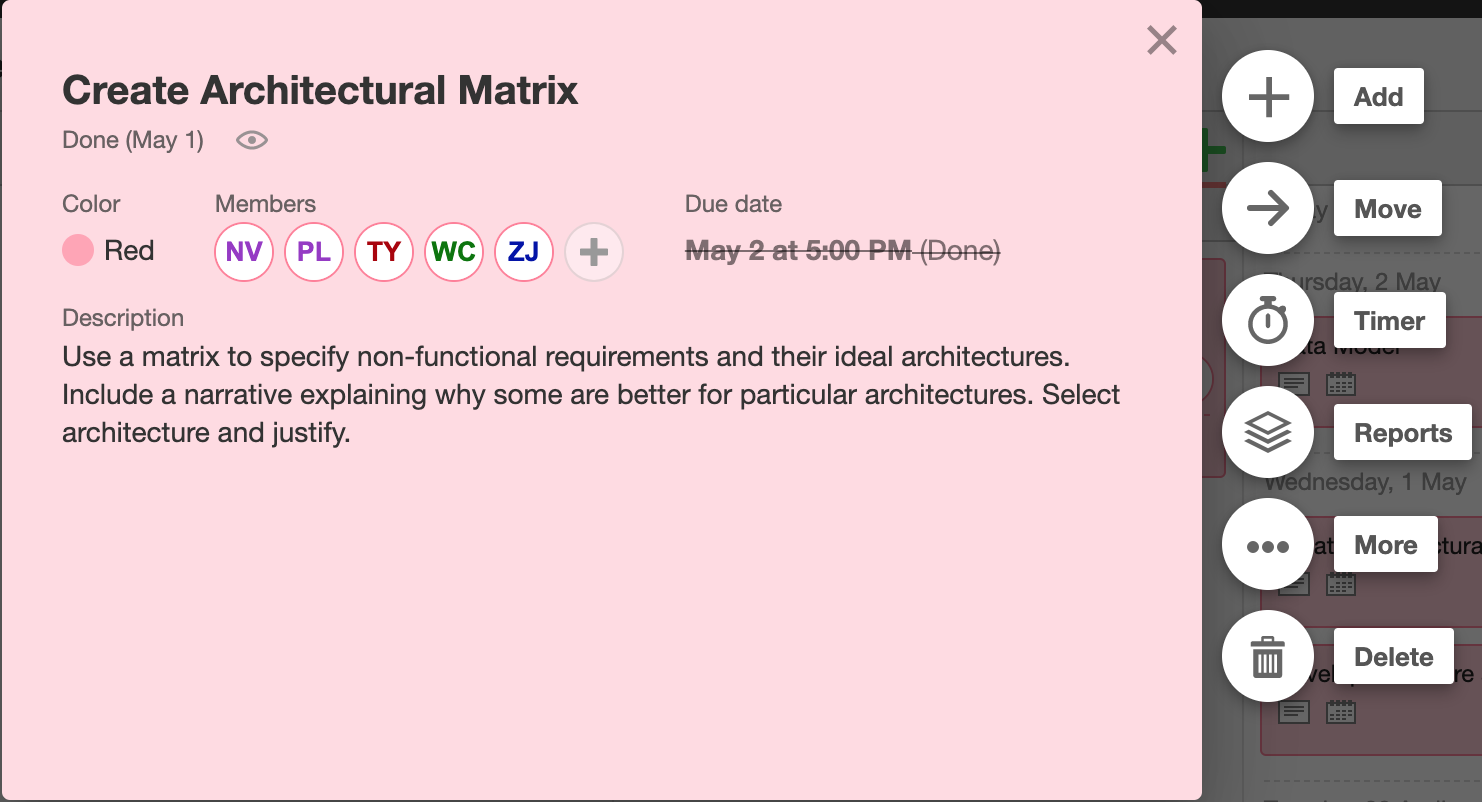
In our updated project plan we have a Gantt chart listing all tasks for each deliverable, to whom has participated in completing them, the start dates, due dates, and completion dates as well. This is so we can effectively negotiate and manage reasonable and achievable deadlines across the team. We color coded our chart with four different colors. Ideally you have two main colors representing “in progress” dates and a “completed” date. Since we have one deliverable left (five) we decided to create two more colors that help show the estimated timelines on our Gantt chart. We created a color for when we “anticipate progress” and also an estimated “project completion” date. Regarding the timeline/dates, the dates that class will be held in the future or already has been held were used.

**Kanban**

**Overall KanBan Board (Red is Deliverable 4):**

****

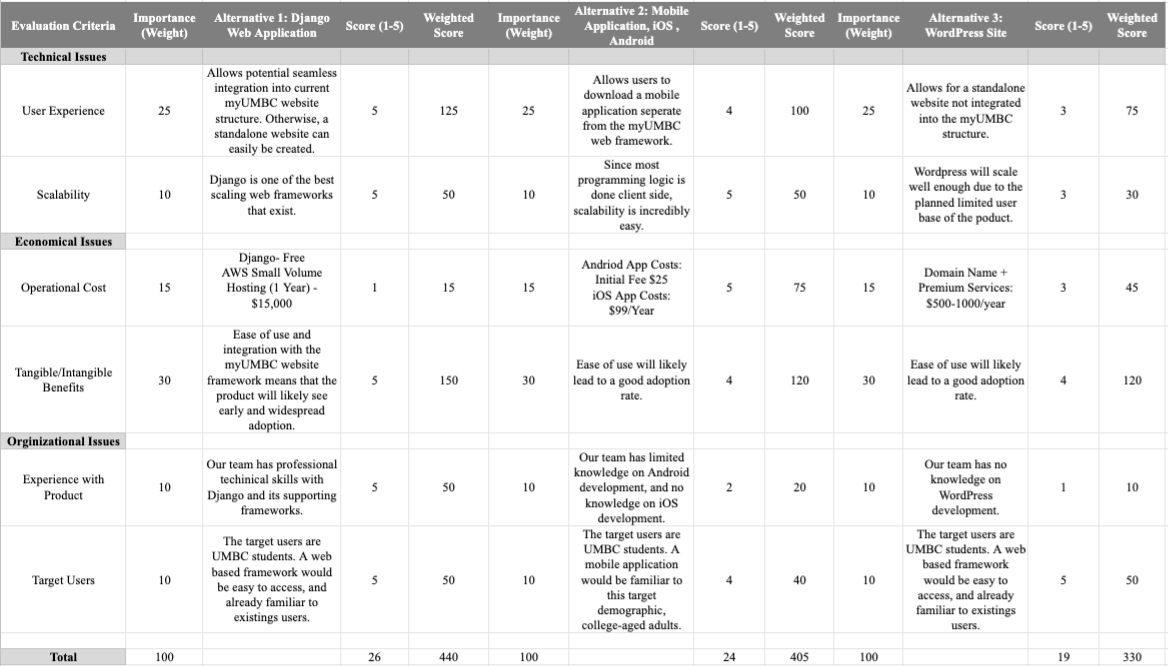
**Individual Assignment Description Example:**

****

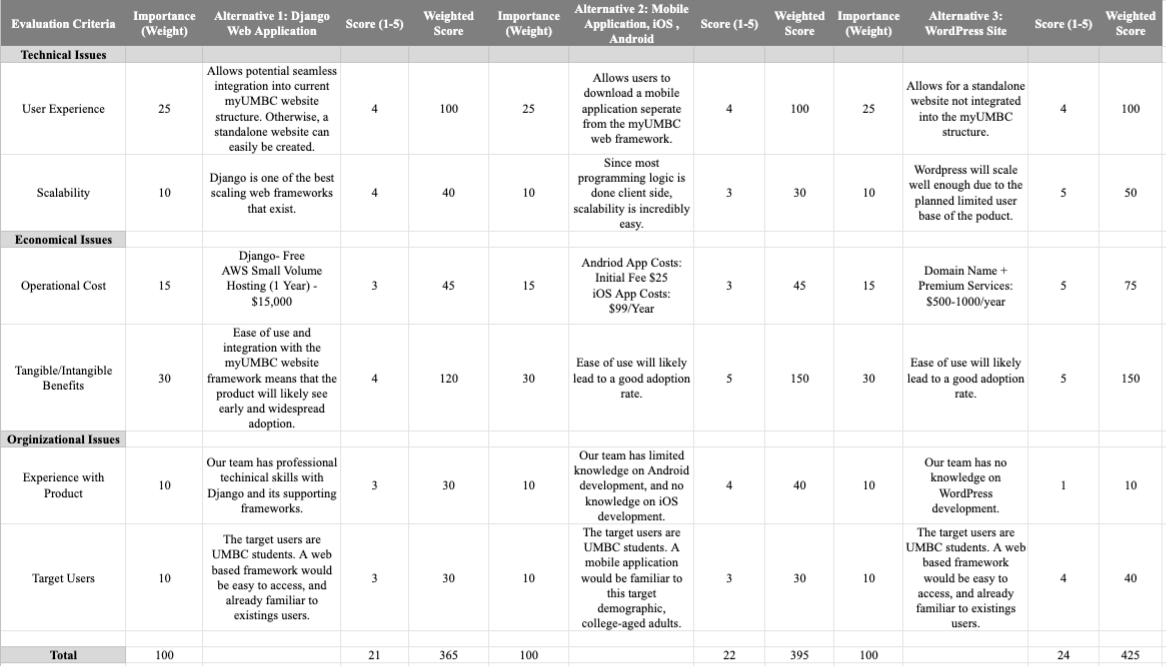
**Appendix**

Individual Alternative Matrices:  **Page 11 - 14**

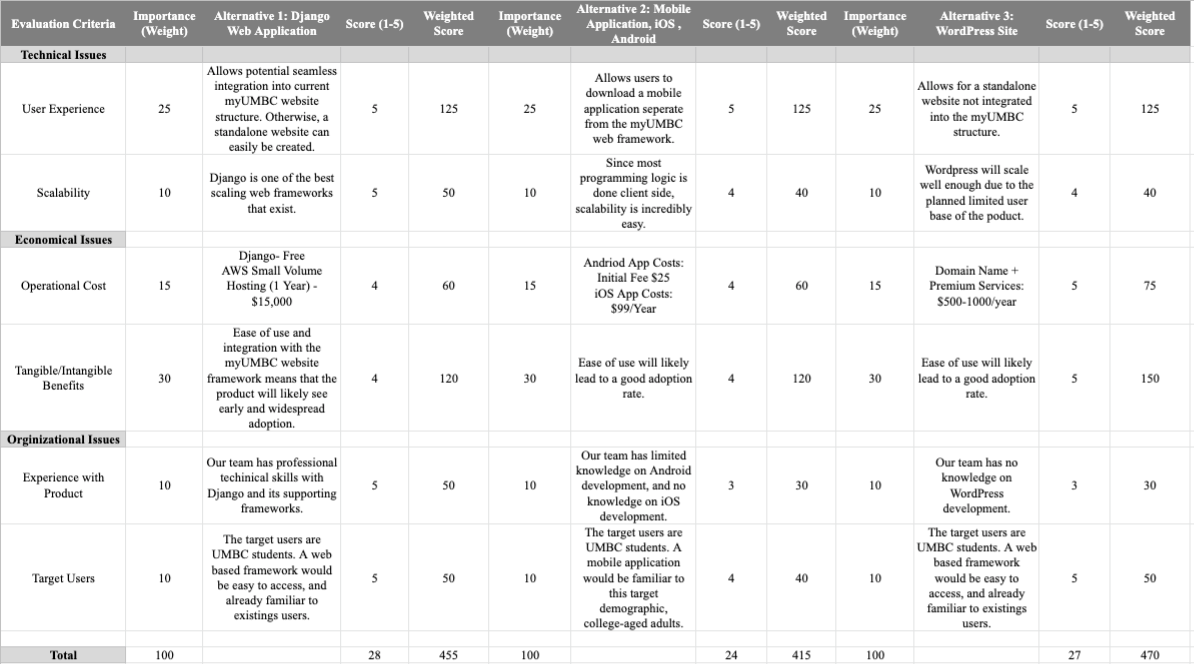
**Paul’s Alternative Matrix:**



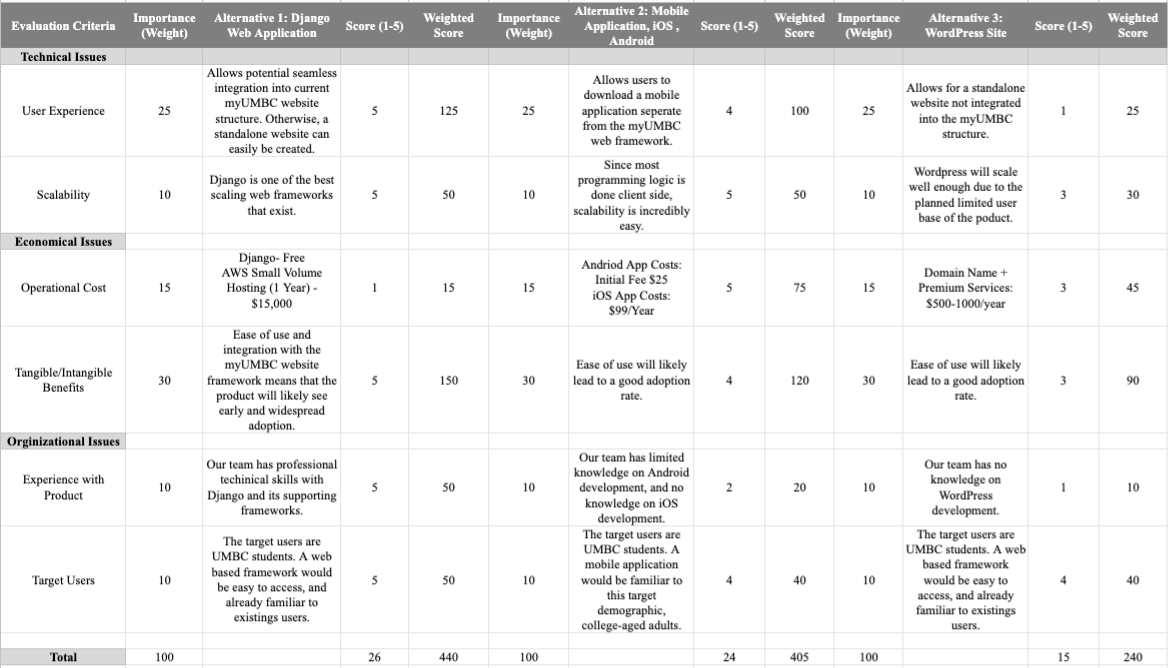
**Ben’s Alternative Matrix:**



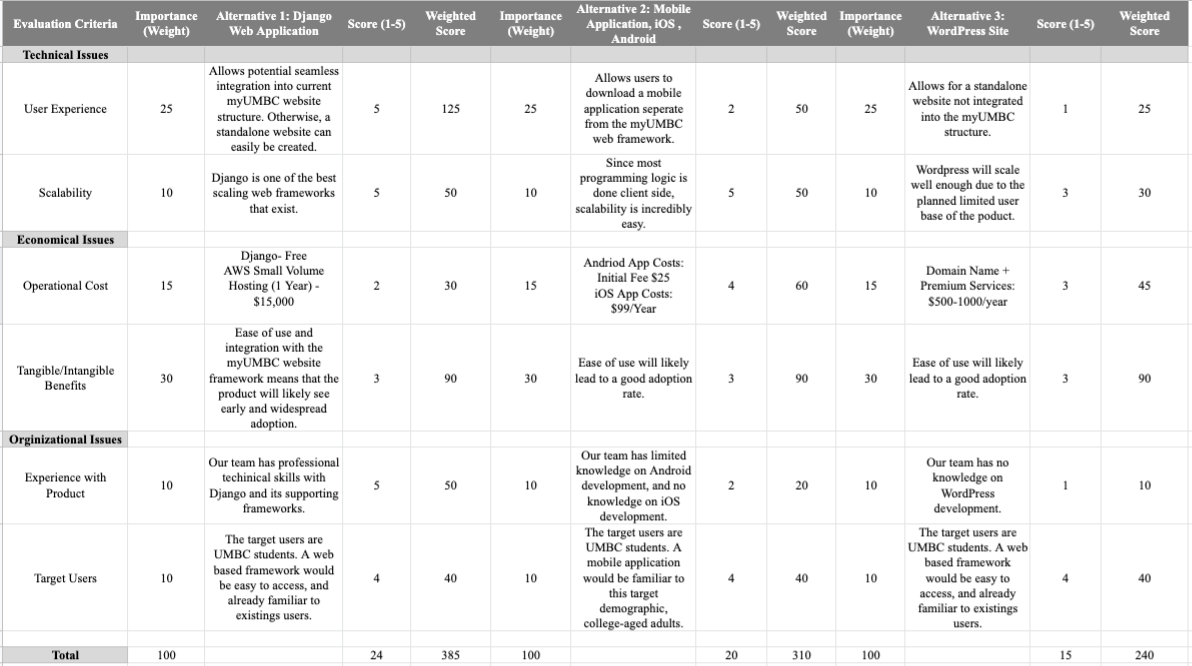
**Densu’s Alternative Matrix**



**Jas’s Alternative Matrix:**

****

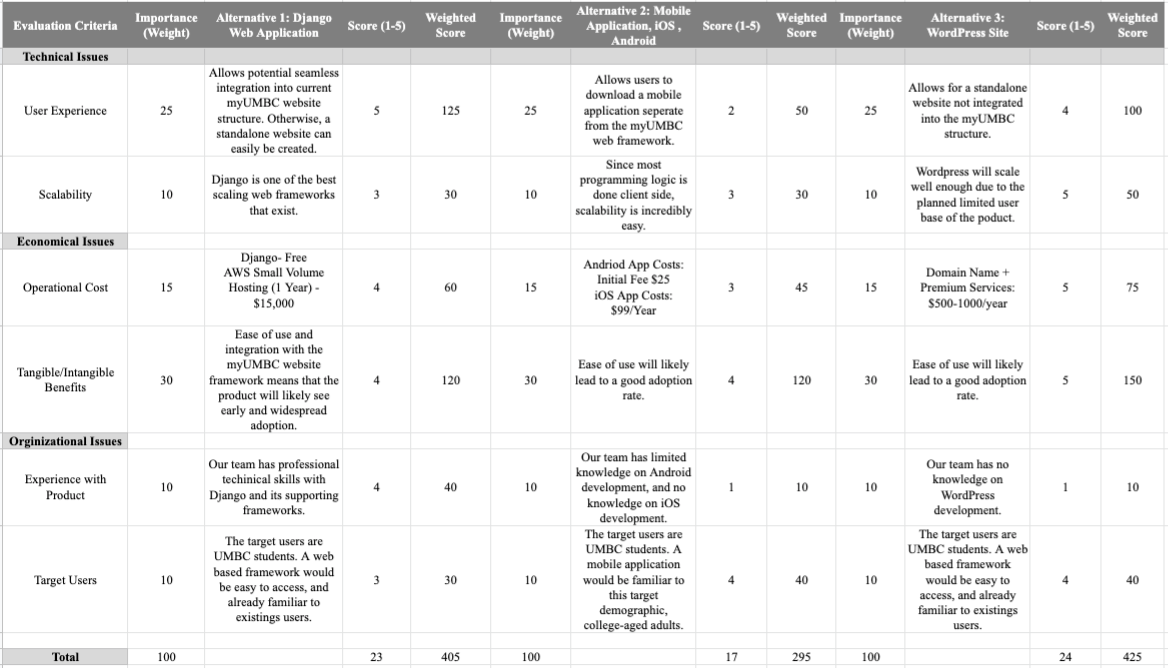
**Tanner’s Alternative Matrix:**

****

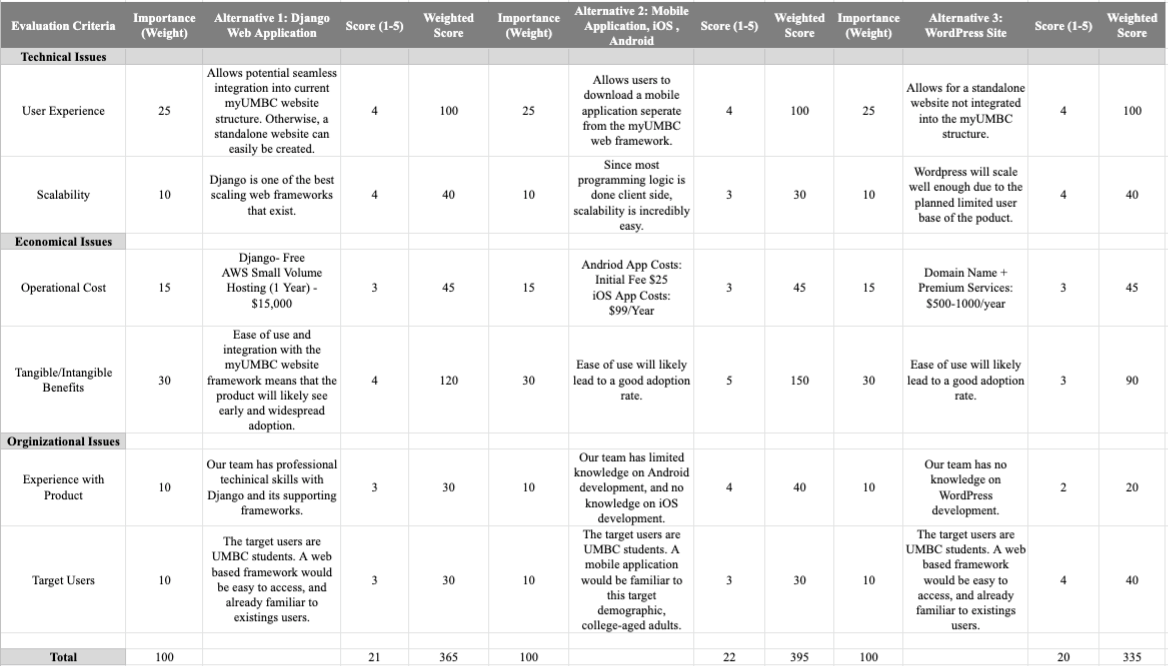
**Neil’s Alternative Matrix:**

****

**Will’s Alternative Matrix:**

****

**Zach’s Alternative Matrix:**

****