

## **Project report for UWMNow website**

**Course:** Intelligent user interfaces CS-743 / CS-657

**Instructor:** Prof. McRoy

**Students:** Seyedamirhossein Hesamian, Sean Craig

### **Background:**

The main goal this class project was to create a platform for students to communicate with fellow classmates and see what is happening at UWM campus at a glance. As we know, due to the time constraints, almost in all of the classes there is no time for constructive discussions and possibly students are afraid of asking questions. In addition, discussion component of D2L (which is a central hub for students and staff) are generally not being used by students as they are afraid of being judged. There is no doubt that having a truly constructive discussion with fellow classmates without being afraid of being judged can result in more effective learning experience and more engagement with studies. It is fact that being more engaged with classes and fellow students, means more likelihood of more study time and being aware of the situation. These factors can result in higher grades and successful learning experience, which is the essential idea behind this class project.

### **Development:**

To achieve the above goals, and proving that creating such a useful platform in a short period time is possible, we used rapid development approach. That being said, we had no choice but to sacrifice best coding practices in software engineering (i.e. time complexity, efficiency, dynamic programming and more) over a working system. On the other hand, we wanted to create a starting point just in case that our application would be used in real world environment. Thus, we followed modular design and we used ORM (object relational mapping), and thanks to this design decision, we created a cluster ready system. The components of our system includes: Node.js which is a event-driven I/O server-side JavaScript environment based on V8 engine which is the same engine being used in google chrome engine, Express.js which is a very minimal RESTful web framework for Node.js and for ORM we used Sequelize.js to generate SQL queries which itself supports dialects including: SQLite, MySQL, Mariadb, Postgresql and Microsoft SQL Server. To make the development convenient, we used SQLite, but we can easily switch the dialect as ORM supports almost any other relational database. In short, we used rapid development approach to create a flexible system in a time span of less than a month.

### **Evaluation criteria:**

To evaluate how closely we achieved those goals we used a web evaluation rubric that graded: affiliation, audience, authority, content, design, learning process, objectivity, and purpose. These were scored individually on a 1 - 4 scale, with 1 being the worst and 4 being the best. We had the evaluations take place in two phases, the first with ten students from a separate class and the second with four students from this class. After the first phase we trimmed down the number of questions once we realized that one of the questions was not very applicable for this project.

### **Evaluation result:**

In the first phase, of the evaluation, only 7 out of the 10 were able to be used 3 were disqualified for either incompleteness. Since we used a rubric that asked eight questions each with a possible score of

1 - 4, for this phase, the site received an average score of 30.6 out of 32 which is equal to 95 percent. These results are in line with what we expected, since the evaluators were students in a class that Amir is a teacher assistant for. In order to have a more unbiased group evaluate the project we had four students from this class evaluate it, for the second phase. We trimmed the number of questions to seven this time and received an average score of 25.5 out of 28 which is equal to 91 percent. With these two sets of results being so close to each other we were able to move to the parts that suggested to continue to work on.

### **Future work:**

The evaluations gave us interesting ways of expanding this platform for the future. Many users especially liked the class schedule evaluator, some of the suggestions were focused on organizing how classes are displayed such as “add a sort function for rating the classes”. Others wanted more information from the people rating classes like, “add the ability to add a description as to why you rated the class difficulty that would be cool”. We also received ideas on how to better integrate parts of the website together such as, “tips to help other students succeed in that class would be a good addition”. Lastly, we received a suggestion that would be out of our current scope but was an interesting idea none the less “homework tracker, input all of their homework and upcoming tests and order them by closest date or highest importance”. Some ways that we could address these suggestions are, to add an optional comment section to the class rating page where the user is able to justify the rating given. Also we can include a link to introduction videos on how to use the website in the activation email.

### **Conclusion:**

We created a highly flexible working system in a short period of time, to address the ineffectiveness of class discussions, thus pushing for a successful learning experience. Following that, we came up with an evaluation rubrics and used it to evaluate the usability of the system. After carefully examining the evaluations and feedback, we found that with a short array of modifications we can address most of the suggestions mentioned in evaluation forms. Therefore, it is safe to that we fulfilled our initial goals and created a positively evaluated application which is a perfect starting point to be used in a real world environment.