

CS315: Group 08

Personal Scheduler

Mohit Kumar Garg	Prithvi Sharma	Sahil Solanki
11431	11541	11624
mohitkg@iitk.ac.in	prithvis@iitk.ac.in	ssolanki@iitk.ac.in
Dept. of CSE	Dept. of CSE	Dept. of CSE

Indian Institute of Technology, Kanpur

Final report
12th April, 2014

Abstract

For a quarter of a century, the relational database (RDBMS) has been the dominant model for database management. But, today, non-relational “NoSQL” databases are gaining mindshare as an alternative model for database management. This project was undertaken to develop a practical understanding of the basics of NoSQL, particularly mongoDB, by creating a simple web application ‘Personal Scheduler’.

1 Introduction and Problem Statement

The problem statement is to get acquainted with the MongoDB framework and create a scheduler application in the process.

We have created a web application aimed at university students and professors. The scheduler allows the users to list out their regular academic and personal schedules in a structured way with some other added functionalities. This will be helpful for effective time management. The academic schedule of a student is determined in the beginning of the semester and remains the same for the whole semester. In order to avoid adding the repetitive schedule of classes every week, a student can mention the courses he/she is attending and they will be automatically added in their schedule for the whole semester.

Sometimes an instructor needs to take extra classes apart from the regular classes and generally it becomes a

headache for the instructor to come up with a time which is conducive for all the students. Having this motivation in mind, we develop this application to provide an efficient tool for the instructor so that he need not to waste the valuable class hours in discussing the timing of extra class.

1.1 Related Material

Technologies used:- MongoDB, HTML, CSS, JavaScript, jQuery

Frameworks and Libraries used:- Twitter Bootstrap, Arshaw FullCalendar jQuery plugin, PHP Driver for MongoDB

2 Database Description

In this application we maintain 4 collections.

1. **Users** : This is to keep the details of all the registered users like username, password, list of courses.
2. **Courses** : This is to keep the details of the courses that are currently offered. This stores course name, instructor name, day and timing of class, students registered in that course, scheduled extra classes. Instructor can add a course in this collection.
3. **Personal Schedule** : For every user this stores the non-academic activities. A user can add an activity with date, time and description.

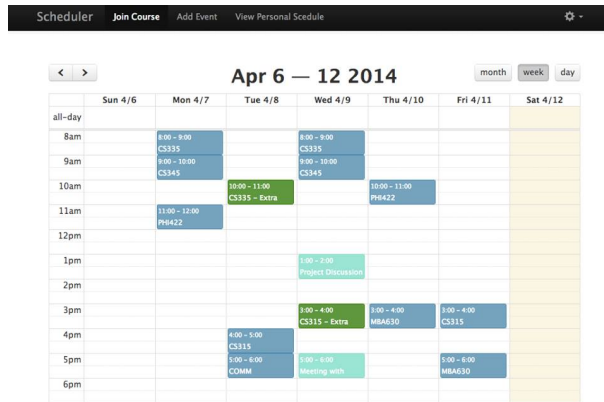


Figure 1: Weekly Schedule

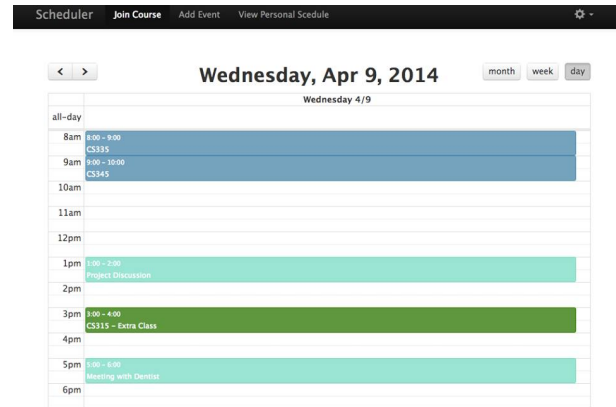


Figure 2: Daily Schedule

3 Algorithm or Approach

1. **Populate the activities in the calendar** : For a user iterate through all the courses he/she is taking, for each course query the regular and extra-class schedule from *Courses* and populate it in the calendar. Then query the personal schedule from *Personal Schedule* and populate this too.
2. **Scheduling Extra Class** : Let say the instructor wants to schedule an extra class of course *c*. *student-list(c)* is the list of students taking the course *c*. Let *CoursesUnion(c)* be the union of all the courses taken by the users in *student-list(c)*. Now iterate through *CoursesUnion(c)* and check if that course has any regular/extra class on the time specified by instructor.

Use the following format for figures:

4 Results

Details of results, in tabular and/or graphical formats.
More importantly, analyze the results.

5 Conclusions

During the course of development of the web application, following properties of MongoDB were observed:

Positives

1. **Schemaless** - As MongoDB is a document data store, it is schemaless in nature and thus it is ex-

tremely easy to add new fields or completely change the structure of a model.

2. **Query Language** - Querying into documents is very easy due to the JSON document storage model. Features can be built quickly with advanced queries.
3. **Fully-featured Drivers** - Official MongoDB drivers are available for all major industry languages.

Negatives

Specific to our relatively simple project, there is nothing in which MongoDB lacks in comparison to relational databases.

But, on further research about how MongoDB databases are implemented at the lowest level, we found some issues people have faced with MongoDB of which a prominent one is the following:-

1. **Uncompressed field names** - MongoDB supports arbitrary documents in a collection. Now, because most of the field names are similar, if one stores 1000 documents with the key "name", "name" is stored 1000 times in the data set instead of shortening the field name or linking all same key names to a location which seems an overkill.

References

- [1] <http://docs.mongodb.org/manual/>
- [2] <http://arshaw.com/fullcalendar/>
- [3] <http://getbootstrap.com/>