Final Report for @10Dance

Group B4

Brandon Maxwell, Curtis Ullerich, Yifei Zhu, and Todd Wegter

Version	Date	Author	Change
0.1	12/02/07	SM	Initial Document
0.2	04/24/12	BM	Added Reflection
0.3	04/24/12	TW	Added Reflection
0.4	04/24/12	YZ	Added Reflection
0.5	04/24/12	CU	Added Reflection
1.0	04/25/12	TW	Final Revision Ready for Submission

Table of Contents

Time for reflection! The purpose of this document is to capture student experiences – both positive and negative – during the semester long team project. What are the key events that are to remembered? What are the key lessons that you learned that you can carry over to future projects. Think back starting from the time the teams were assigned, then project proposals were made, screenshots developed, requirements, architecture, framework demo, test plan, and finally the demo. What would you do different if you were assigned the same team and the same project again?

Each team member is to record his/her thoughts on a separate page. Thanks!

1 Team Member 1: Brandon Maxwell

1.1 WHAT WENT WRONG

I think we only went wrong with this project because we didn't start with any knowledge about any of the ideas we used in the project. My job was to get the database working and since I had never done anything with databases I tried to do things that someone with experience wouldn't have done. The same thing happened with how we did interactions with the user on the JSP pages, since we had never worked with them we didn't know about or how to use specific software that would have solved some of the problems that we ran into.

1.2 WHAT WENT RIGHT

We started programming right away so when it got down to the end of the semester we didn't have to panic and try to come up with something quickly. This allowed us to meet with the TAs and show them something useful so we had a clear idea of whether what we were doing was correct for the course or not. Also, we split up the work very well so we all always had something to do so no time was wasted.

1.3 LESSONS LEARNED

I learned a lot of things from this project based on the fact that I didn't know anything about the languages or ideas that we used in the project. This includes: working with databases, Javascript, Html, how to use JSP pages, and running Servers. I also got more experience working with groups to finish a project.

2 Team Member 2: Curtis Ullerich

2.1 WHAT WENT WRONG

We had to learn pretty much everything about this project (aside from Java) from the start: HTML, Javascript, JSPs, Servlets, databases, and how to use all of them for session management and client-server interaction. We had a couple of design mistakes due to constraints of using JPA for our database storage with Google App Engine. We weren't able to use polymorphism or inheritance like we had planned so our code structure got messier than I would have liked.

2.2 WHAT WENT RIGHT

For the most part, we worked well as a team, learned a lot, and cranked out a good project. When we met, we were able to work cohesively and efficiently. We got good feedback from our clients and were able to work adaptively to suit their needs. We split our work into domain-specific areas of expertise and became experts at those tools (within the scope of our project). This was a great way to parallelize our work.

2.3 LESSONS LEARNED

I think that our processes were very good, overall. I did learn some strategies for dealing with difficult team members and how to mitigate weaknesses of individual contributors. I've learned that I need to reevaluate how I use databases to minimize interactions with it; that I need to learn JQuery; that interfaces should be clearly defined with maintainability and *extensibility* in mind; and that sometimes the best way to learn a new technology is to meet your goal any way possible, and then evaluate your methods and do it better.

3 Team Member 3: Yifei Zhu

3.1 WHAT WENT WRONG

- We should do more research about how Google App Engine stores data and have a better idea about how Google App Engine works, so that we might have avoided being unable to store object in Google App Engine.
- We spent most of time coding during the meetings, and asked each other questions while coding. It is good because we can help and communicate with each other right the way, but it might be more productive if we can spend more time working alone and have less time in group meetings.
- I wasn't very talkative and outgoing enough in the team meetings, since I do not have that much experience working with all native speakers for a big project. I didn't understand the conversation sometimes, and I didn't have enough communication with team members, it may have lead to misunderstanding sometimes.

3.2 WHAT WENT RIGHT

- As a team, we were able to meet most Friday nights and Saturday afternoons and worked hard on the project.
- We were able to finish all the assignments either early or on time, and met with the TA if we have problems.

3.3 LESSONS LEARNED

- Learned more about how to write good test code.
- Communication is really important in team work; I should be more talkative and outgoing.
- Learned how to work well in a team.
- Learned about the product lifecycle to how to create a product.
- Learn how use the Google App engine and some other new languages.

4 Team Member 4: Todd Wegter

4.1 WHAT WENT WRONG

In this project, not many things went terribly wrong. The project involved concepts that were novel to all of us, so there was a very steep learning curve. Two main hitches occurred in the creation of our attendance system. The first problem we discovered was that we could not have objects as parameters for something being stored in the Google App Engine's Datastore. This surprised us and required us to do a complete paradigm shift in how we were storing things in the datastore. Instead of storing objects as parameters for an object, instances of an object stored references to other objects in the database. This added complexity to the inner workings of the project.

The other major hitch we encountered was that the datastore could seemingly not handle polymorphism or inheritance in the objects we were storing in the datastore. The datastore would sometimes store objects as the actual type, and sometimes it would store them as the parent type. Then when we would try to find all instances of a certain object in the datastore, the query would not return all of the entries. We could not diagnose this issue or find any other information regarding similar issues, so we solved the problem by flattening our Java hierarchy. This prevented the datastore from becoming "confused" over the type of an object.

As a team, we worked well together for the most part. One team member didn't do as much work, which caused frustration among other members. It seemed as though that member didn't understand that we had a deadline to meet and that we all needed to push towards the end to get the project done. Other than that, however, we worked well together and had very productive meetings.

4.2 WHAT WENT RIGHT

One of the best parts of our group was its commitment to the project. On the whole, we were very dedicated to the success of the project, and we were willing to do whatever it took to see that the project would succeed. We started coding within the first couple weeks of the project, which greatly enhanced our ability to have a working project to demo at the end of the semester. All of our team members put in a large number of hours (some more than others) every week to ensure the success of the project.

Our decision to host our app through the Google App Engine also turned out to be a good decision despite the limitations of its datastore. At least one of these limitations could have been anticipated if any of us had had any experience with databases, so that issue wasn't really GAE's fault. The free hosting that GAE provides and the way it manages the app will be great for actually using this application with ISU's marching band. The features provided by GAE will make maintaining the system much easier (and cheaper) for the directors of the marching band.

4.3 LESSONS LEARNED

I learned that starting early is the best thing you can do in a project. I already had an understanding of this concept, but this project really drove it home. I also learned that choosing a good team is everything. I knew two of my three teammates before the project, and I knew that they would work hard. This turned out to be a huge advantage. Since I had handpicked teammates that would work as hard as me, I wasn't the only one pushing to start early and get things done. This made for a much less stressful environment.

On a less philosophical note, I learned a ton about web development. This is great knowledge to have under my belt. I am sure that I will use this knowledge multiple times in the future. I also learned that software documentation is a lot more effective when completed before the software it's outlining is completed.