Final Report for <<CyDisc>>

Group SB\_C\_4

David Kirshenbaum

Brian Newman

Pierce Adajar

Zhanghao Wen

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change** |
| 0.1 | 12/02/07 | SM | Initial Document |
|  |  |  |  |
| 0.3 | 17/12/08 | Zhanghao | Team member’s reflection + format |
| 0.4 | 12/08/17 | Bjnewman | Reflection |
| 0.5 | 12/08/17 | Pierce | Reflection |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Team Member 1: David Kirshenbaum 4](#_Toc164665234)

[1.1 What went Wrong 4](#_Toc164665235)

[1.2 What went RiGHT 4](#_Toc164665236)

[1.3 Lessons learnt 4](#_Toc164665237)

[2 Team Member 2: Brian Newman 5](#_Toc164665238)

[2.1 What went Wrong 5](#_Toc164665239)

[2.2 What went RiGHT 5](#_Toc164665240)

[2.3 Lessons learnt 5](#_Toc164665241)

[3 Team Member 3: Pierce Adajar 6](#_Toc164665242)

[3.1 What went Wrong 6](#_Toc164665243)

[3.2 What went RiGHT 6](#_Toc164665244)

[3.3 Lessons learnt 6](#_Toc164665245)

[4 Team Member 4: Zhanghao Wen 7](#_Toc164665246)

[4.1 What went Wrong 7](#_Toc164665247)

[4.2 What went RiGHT 7](#_Toc164665248)

[4.3 Lessons learnt 7](#_Toc164665249)

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 learnt 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!

FOCUS ON LESSONS RELATED TO

1. Working in a TEAM
2. Software Development processes
3. Technical Issues

# Team Member 1: David Kirshenbaum

## What went Wrong

The main thing that went wrong is we could not accomplish everything we wanted to. There were a few reasons, one being that many of us would not be able to start as early as we’d like to. Also, with people being busy with stuff in other classes, sometimes it was just hard to get started. Lastly, some problems for me were not understanding some of the innerworking of android and how the app really communicates with itself.

## What went RiGHT

Our team was succesfully able to delegate responsibilities. Every demo everyone had their objectives and completed them. This was written out using resources like slack and trello and allowed us to work without fear of wasting work or not being aware of what was expected from each other’s code. We also managed to follow good design patterns for the most part. There would be classes for dealing with input/ouput, one for processing, and one for communicating with the server. We also made sure everyone learned, because everyone worked in some capacity with android, server code, and database management.

## Lessons learnt

Of course the main lesson is to start early. The ideal way to approach this class would be about two hours a day, but sometimes the motivation or time isn’t there. Sometimes working with a team also slows down the process, and it is important to say this person has to do this, by this day, otherwise you get bogged down talking about how to do something and not actually coding. Lastly, for me personally, you need to have good knowledge of the software you are writing with or there will be sloppy code. It wasn’t until a few months that I really started to understand some of the innerworkings if android (and I’m still shaky to be honest), however, I did manage to create my fragments and they work pretty well. Were I to have another month, I could accomplish a ton more, but I am glad I did get to learn android/java.

# Team Member 2: Brian Newman

## What went Wrong

I believe the main thing that went wrong was that we didn’t manage our time adequately and did not finish all the use-cases we planned on. The main cause of this issue for me was developing an android application for the first time and learning networking between multiple devices. Time management was an issue, since at times we would become busy with work, other classes, etc. where we wouldn’t have put enough time as we should have. Lastly, integrating our entire project as a whole took time to complete and was more complicated than I thought.

## What went RiGHT

I believe communication went well and designating which tasks went to who. For example, if there were issues with IDE/project setup, we would talk to Pierce, or if someone needed assistance with the database they could contact me. Therefore, roles and tasks were established and accomplished for the most part by each team member.

## Lessons learnt

I believe the biggest lessons I learned from this team project experience was the importance of time management and integrating all the parts of the project. For example, when we combined our projects there were a couple differences in the login part of it. So, when we tried to compile our login had broken so we had to fix it. That is just a small example of the issues we ran into. Lastly, time management was a big issue for our project. In the future I need to plan my time and tasks better to fulfill all the use-cases that should have been implemented. I also learned from this that team projects can be very productive, and a lot can be accomplished in a timely manner with proper team work and attitude.

# Team Member 3: Pierce Adajar

## What went Wrong

For me, the two major things that went wrong were 1) not loving the project idea 100% and 2) poor communication, especially regarding git. Personally, I think I had a hard time motivating myself to work to the extent of my ability because, just like we were warned at the beginning of the semester, it’s hard to dedicate a significant amount of time and effort to something you are apathetic about. Secondly, we ran into a decent amount of issues, especially later on in the semester, regarding git usage and practices. I think a lot of it could have been avoided if we had set up clearer regulations, guidelines, and expectations about git at the beginning of the semester, but unfortunately, we did not have the foresight to do so.

## What went RiGHT

Overall, I am very proud at the way my team did handle miscommunications and misunderstandings – I think that really helped us solve a lot of our issues we ran into in a timely manner. Additionally, I think that everyone did a really great job of not only completing the tasks they were assigned, but also having the initiative to step up and say, “this is something I’m good at,” and taking on tasks that they brought their A-game to – ultimately making a very valuable contribution to the team. Having a wide range of skills across our members really helped when someone ran into an issue they didn’t know how to solve, because they were able to reach out to someone in our group and ask for help.

## Lessons learnt

In the future, set up a framework not only for the project, but also for the management / codebase much earlier on. I really enjoyed using Trello, and is certainly something I will be using for my own projects in the future. When I work in a group, I will make sure that everyone is on-track for git usage, as well as possibly pre-configuring project branches ahead of time and use a more hierarchical system for who is allowed to push where, who can add/drop branches, etc.

Another big thing is emphasizing smaller commits more often – I myself am guilty of this as well, but definitely leading by example would help in the future. Being able to consistently show progress as well as running into fewer merge issues is a huge deal.

# Team Member 4: <<Zhanghao Wen>>

## What went Wrong

1. I should spend more time on learning new programing language and coding. Understand basic theory and concept before doing projects rather than doing and learning the project at the same time.
2. Got stuck and waste too much time on certain debugging problem.
3. Did not finish all the features that was planned to finish.
4. Did not have time to polish screens to make it looks like a actual application.

## What went RiGHT

1. Understand the general idea of how the whole projects working.
2. learned android(java), server(mainly php file), database, and how those things inter-connected.
3. Used fragment to integrate through whole projects.

## Lessons learnt

1. Keep progress ordered and cleared (management).
2. Start doing things early, start now.
3. Seek help frequently, take advantage of TAs
4. Apply theory -- what we learned in class to practice – doing the projects.