Michael Fiddes, William Papsco, Ashley Maddix, Zayne Stites, Colton Russell, Marina Neumann

Term Project: *ChocAn*

Final Retrospective

Table of Contents

1 Introduction **2**
1.2 Target Audience 2

2 Successes **3**
2.1 Transition from Design to Implementation 4

3 Failures **4**
3.1 Database Security 4

4 Needed Improvements **4**
4.1 Technological Difficulties 5
4.2 New Group Dynamic 5

5 Conclusion **5**
5.1 What We Learned 5
5.2 For Next Time 6

1 Introduction

At ICNH, we aspire to build creative solutions to complex problems and for ChocAn we hope that we've provided the best software to help manage and maintain the health services for those who suffer from Chocolate Addiction. Our goal was to create an application that could be used by both health providers and managers at ChocAn to record health services, manage members information, track patient/provider history and generate provider and member reports to managers. The services the ChocAn provide help millions of chocolate addicts every year and so we built an application that was reliable and easy to use for all parties involved. In the next pages we will evaluate the strengths and weaknesses of our final product, as well as critique the development process and reflect on where improvements can be made, and where code needs to be managed.

1.1 Purpose and Scope

This document is to reflect on the completed software product delivered to ChocAn and critique where the software could benefit from improvement or redesign, as well as important aspects that will need to be managed moving forward. We will discuss the good and bad of our development process, as well as critique our final product. There is no such thing as a perfect product, and while we know our product is imperfect, we will discuss how we delivered on our requirements for the system.

1.2 Target Audience

This reflection of our recently developed software is written mainly for our engineers to consider the different aspects of development and discuss several improvements to be made if ChocAn continues

to employ our services. As designers and engineers we are always learning, and discussing what worked, what didn't, and how we can improve which will ultimately help us grow as developers.

2 Successes

As with any project, it is critical that we not only reflect on how we can do better next time, but also on the strategies and methods that worked well for us. Understanding and categorizing these successes is useful for putting ourselves on the fast track to success in future projects. There were several areas that we felt best highlighted our success at meeting the software requirements.

2.1 Transition from Design to Implementation

As a group, we spent a lot of time on designing the structure of this software and figuring out how to best configure a solution that met ChocAn needs and requirements. Implementation began near the end of the design phase when we felt we had a firm understanding on how to proceed in development. Having a clear outline of the flow of data and control between subsystems allowed us to very quickly develop a high level framework for the software that kept us from running into any major road-blocks during implementation.

2.2 Database

Initially, some of the group members had been planning on using some simple CSV files to store the ChocAn data. One group member, however, strongly recommended that we use a real database and volunteered to set one up with MySQL. This proved to be an excellent decision as it provided great flexibility in how we could interface with our data. Primarily, it made the software more modularized and

therefore easier to access and expand in development.. By simply having a class dedicated to helper functions that interface with the database, we were able to centralize all accesses to the data. This is a much better alternative than having to handle the opening and closing of CSVs all over our code and keep track of file streams. Secondly, if we needed to interact with the database in some way that we didn't have the ability to, providing that functionality was as simple as writing a new helper function in the class that handles the database. On multiple occasions we needed to expand what we could do with the database, so we simply described the problem to the group member working on the database, and often 10 minutes later had that functionality available for use.

3 Failures

3.1 Database Security

After careful consideration, we have come to the majority conclusion that using the password "password123" was a poor decision for securing our database.We had initially pushed everything to github and hadn't thought about the potential consequences to having hard coded the password in our database interface file. We suspect this vulnerability is what led to our database being stolen and held ransom. After recreating the database, we employed a much more complex password and secured access in the hope of deterring future attacks, particularly by moving the password to a separate source file not uploaded to github.

4 Needed Improvements

4.1 Technological Difficulties

While our work in the design stage set us up with a clear vision for the implementation phase of development, we experienced significant slowdown when we actually began writing code as a result of our circumstance at the time. We made the decision to use github for our source control, and to write the software in Java using IntelliJ as our IDE. Unfortunately, only two people in the group had experience with any form of source control, and only one member of the group had any meaningful amount of experience in Java. Some members of the group had little to no experience using an IDE as well, so there was a fair amount of overhead associated with early development of the software. We still feel that these choices were beneficial overall, but in future projects we definitely need to focus on understanding how to use the tools we plan to use, before we actually start using them.

4.2 New Group Dynamic

This project was the first that our group developers worked together on. As such, much time was spent on learning how individuals work best and what each individual is skilled in. Some of our group members produced their best work long ahead of deadlines, others needed the pressure of an imminent deadline to encourage them. In addition, it was difficult for some of our members to work on collective code after having spent years working alone. Once we figured these things out, it was smooth sailing through the project.

5 Conclusion

5.1 What We Learned

This project was a dive into the world of software engineering as a team. The experience of collectively designing and building a piece of software is very different from working on a project alone. Through this, many of us were able to expand our experience to include source control and navigating IDEs, but we also learned how important it is to efficiently utilize each individual's strengths and weaknesses to progress. Every person in the group was able to take over some portion of the project that they worked best on, whether that was the database, the documents, implementation, or testing. While there were many cases where people needed to do some work in other parts of the project, each person having that overarching focus on just one piece of the puzzle worked out very well for us. One of our greatest strengths as a group was continuous communication and collaboration through any compilation errors or design reconsiderations.

5.2 For Next Time

With our new understanding of the group dynamic, and newly acquired skills with github and IntelliJ, the next project will run much smoother.