Group 47 - Contribution Report

Christian Onishile – 965506

During the implementation period I was responsible for the production of the backend code involving the classes I designed in assignment 1. As such, I was responsible for writing code involving the librarian and user accounts. In addition, I also was responsible for writing the code involving operations on the account's profile image. As such I had to write the code that allowed users to choose profile images and draw new profile images. The java class and fxml classes I made were subject to much change and amendments within the integration stage as they were produced in isolation from the other features of the library application. In hindsight it might have been a good idea to utilize more XP and agile principles, such as pair programming or continuous integration, to result in a smoother integration process. Throughout the implementation stage we would use various branches on Github to allow each group member access to the current state of each class and method.

Benjamin Farrington - 915699

For the second assignment, I kept the role of group secretary whilst doing the initial implementation of the DVD class, as I Designed in A1. I also paired with Cristi, to build the GUI for the project, I initiated the GUI Implementation building a way to easily create windows, created the dashboards, completing the User dashboard and its operations then implementing the View Profile and Edit profile windows whilst Cristi focused on the Librarian dashboard and mainly it's operations. Finally in the GUI build, I worked with Christian on adding the draw profile image functionality to the final GUI version, Due to time constraints Chris's design was simplified to reduce error. Limiting things such as drawing tools.

I believe we could have improved as a group by doing more work in the same room, as delegating the tasks we derived from the assignment led to certain features being too system specific or off specification making merging of the features a difficult task. We also went way over our implementation deadline of the 3rd of december, but that deadline may have underestimated the complexity of our design.

Alex Moras - 956156

My role in the project was to start packaging the software in to its final form. I started by assisting Ryan with the design of the database handling using JPA and Hibernate. We then implemented a version of the software that uses SQLite instead of MySQL which removed the requirement for a local server to be installed. When the GUI and Database were completed, I started work on linking the two systems together. On doing this I noticed a few little details missing in the GUI which were then added by Cris and Ben. Any other features that weren't working I attempted to fix myself. I spent a lot of time changing how parts of the GUI are referenced and how these calls work in relation to the database. I had to manually merge Git branches which took a lot of time due to issues on other people's branches.

I feel that the project as a whole started off well. However there were definitely a few issues along the way. I think the biggest issue of all was that the people doing the database didn't understand any of the GUI and the people doing the GUI didn't understand any of the database. This meant that when it came to implement the two, things were tricky as they don't work together well. Further to this, because of the very specific way everything has been designed, it makes it extremely hard for multiple people to work on the linking of the two systems. To put this in perspective, whilst it works on my PC, I can't anything but the final packaged version to run on my laptop. This shows that it's simply to difficult with our design for multiple people finalising things. If we took this in to account to start with, our final project would have all aspects of it working flawlessly.

Ryan Williams – 963420

In this assignment my focus throughout was on the research and implementation of our database systems. Having collectively agreed on using an SQL database to store necessary data files, I began work in the first week on researching appropriate APIs and software to decide which would best suit our needs, deciding on using the ORM tool hibernate along with JPA (java persistence API) I was assisted in this step by Alex. From there I preceded to learn and practice coding with this new software library, eventually beginning to add the basic JPA framework features to all of our core classes so that they could be used to interface with firstly a locally hosted mySQL server. Of these base classes I was the author of Resources.java and Copies.java. Once I was comfortable with the API I began work on the more advanced features of our database design. To make this tricky step easier I created another two additional classes, HibernateUtil.java and request.java. HibernateUtil.java was created to ensure the

easiest use of hibernates features. Request.java was created to support the main transactional features for the system, such as requesting a copy of a resource and the ability to collect a copy of a resource to name a couple of the features. This implementation of all the main ‘backend’ database commands/actions took me a considerable amount of time to create and then debug. Nearing the end of the allotted time, I provided additional help to Alex in his implementation of the GUI with the database ‘backend’, mainly along the lines of suggesting best uses of the hibernate API and the code I had written.

Reflecting on how our group worked together I would say we were a cohesive team with frequent fruitful communication. To our merit we kept all discussions professional and productive. Two issues which I believe slowed our progress was incorrect uses of our GitHub repository, as well as different members implementing different versions of Java and accompanying libraries. We should have all prepared for these issues by standardizing which software we planned to use, as well as all familiarising ourselves better with Git. Another issue I believe we had was evaluating the time needed for certain tasks, this I believe lead to some members overall contributing more than others on some weeks. This was to no fault of any individual member, in hindsight I believe we should have broken tasks down into smaller subtasks and then delegated those, for a more even workload for the members.

Keiran Hughes - 913861

My initial task was creating the Book.java class that was designed during the design phase. This class is used to create and manage book resources in the database. I then took on the task of making the application documentation, this meant going through my the java classes made by my other team members, commenting on the classes and methods and ensuring they were up to standard coding conventions. To ensure the task didn't become overwhelming we decided it was best to perform the documentation throughout the creation of the application. The task was made difficult by the use of GitHub branches members keeping localfiles meaning there were multiple versions of some classes, I think this problem could have been decreased by regularly uploading the classes to each branch. Overall I feel that the assignment went quite well with tasks being split equally and that the whole group worked admirably.

Cristian Neacsu - 964379

First I have started by implementing the java class Laptop for which I have done the CRC card in A1. Then as a team we have devided our work in different sections and we have started doing pair programming. I teamed up with Ben Farrington and we both worked side by side to create the GUIs and implement the code behind them (controllers, windows, dashboards, relations with databases, application's execution).

At the beginning we have started with creating a basic design using the examples provided in the lectures. We have managed to create a login screen, 2 dashboards: one for users and one for librarians, and we made a system that projects any new activity in a new window as an independent one. After that, we have redesigned it in order to manage better the course that our application follows: login -> dashboard(user/librarian) -> activities -> dashboard -> logout -> login.

Afterwards we have decided that we will work on different activities to manage our limited time better. I have created, designed and implemented: Creation of a copy (java/fxml), Creation & Edit of a resource (java/fxml), Creation of a Scene Controller (which stores static and dynamic variables to be referenced) and an Operations class that handles loaning, collecting and returning of a copy.

Although fully functional on our side we needed the database setup, so we both commentated our classes processes in order to set everything up when we meet all toghether.

Meanwhile, I have implemented CSS into our application to give it a better and individual look.

In the end, we didn't expect that people that worked on the database and people that worked on the GUIs had to know more about each other's work in order to be more productive and ease our work. This provoked some minor inconveniences when assembling everything togheter.

We have definetly learned how pair programming and relations between teams work and how to improve for the next time, but mainly we have learned that we need to be aware of everyone's work and changes to the whole project.