Name: Mitchell Hird
Due By: January 18th, 2017
Student Number: 100881078
Course: COMP 4905 - Honours Project

Honour's Project Proposal

Project Supervisor: Jean Pierre Corriveau

Project Overview:

This project would be a Java based, networked application that would be focused around one of the world's most popular card games, Magic The Gathering (MTG). When interacting with this application, users will be able to generate and perform easily modifiable searches that access the large card dataset in a quick and efficient manner. In addition to viewing the large amount of cards that are present across many different sets, the users of the application will be able to archive decks that they have personally created. These decks would fall into one or more of the established archetypes, and upon user submission, these decks will be verified in order ensure that the deck adheres to the numerous deck building restrictions for the given archetype. If no errors are present within their creations, the deck and all of its contents will be stored both locally in a client-side database, and a server application that is located at a separate location. Once stored, other users of the application will be able to connect, view, comment, and share decks. Finally, once a deck is archived in the application, users will be able to perform general statistic operations across the creations they have made. These statistics include calculations in deck colour ratios, the spread of converted mana costs across the deck, details on financial costs of individual cards, and the number of times a card appears across all decks within a given archetype.

Motivation For The Project:
The motivation for this project stems mostly from the large community surrounding the Magic The Gathering card game. For over 20 years, players of this game have constantly been creating new decks and ideas, and they are always looking for new electronic resources to help them during this creative process. Currently there are some pre-existing tools that provide search and financial database functionalities, but, I don't believe there is an existing application that provides Magic users with both of those functionalities, while also giving them the ability to submit and share new deck ideas among others in their group. So, with that in mind, I plan to create an application that both suits those needs, as well as provides a positive user experience to all individuals involved.

Main Objectives:

The primary objective of this application can be easily be broken into three sub-objectives. The first objective is to provide users with a stable application that allows them to perform searches

on the existing Magic The Gathering card system. While searching through this database, users should be able to select cards and move them to either a brand new deck or to an existing deck within the archiving system. The second objective of this project is to provide users with a tool that will allow them to create, save, modify and share decks among a group of individuals. This objective will be accomplished by creating a server application that will act as a central hub for all user creations. Once a user created deck has been generated, it will be uploaded to this server, and all other connected users will be able to browse, view, and comment on the new submission.Finally, the third objective of this project is to supply the MTG community with an application that will allow them run statistics and view detailed information about decks and the archetypes that the decks falls under.

Equipment Requirements:

Here is the list of the equipment that this project will need throughout the course of development:

- Java installed on the machine that wishes to run the application
- An existing Magic The Gathering database that contains all of the current cards. This database portion will be retrieved from an external, pre-existing public source
- Another database that the application will use for internal persistent data storage
- JUnit testing library: This Java library will be used for creating the unit tests, and integration tests outlined in the schedule below.
- GitHub Source Repository: This piece of equipment will be used to store all source and test code within an external location. In addition, this resource will be used to track and monitor the work and amount of changes made throughout the project development cycle

Milestones, Deliverables, And Project Schedule:

Project Aspect	Description	Expected Timeframe
Theory work, code/model planning and use cases	This section of the project is to establish the expected back-end logic, use cases, and rough graphical user interfaces that the project will need throughout its development. During this time, use cases will be drafted in attempt to come up with an established system that will outline future test plans, as well as give proper steps on how to use the application from a normal user standpoint. During this time, no coding work will be completed. This process is solely set for planning what the project will need to do once it is complete	Dec 18th - Dec 24th
Implementation Of Core Project Functionality	During this portion of the project, the core code functionality will be implemented. This functionality includes implementing the necessary code for user	Dec 24th - Jan 24th

	controlled database lookups, local storage deck submission and deck archetype-based rules validation. At this point in time, the application will solely be a single application that stores all information in a persistent, local, database.	
Testing Of Core Project Functionality **(Deliverable Due - Feb 14th)**	This portion of the project will be the part where core functionality tests are created. These tests will be normal unit tests written using the well-established JUnit test framework, and they will be used to ensure that core functionality (database queries, and proper deck rules validation) is working as intended. Once these tests are completed, they will continuously be run against the project once any new work is completed. Doing so will outline the functionality that broke between implementations, and what needs to be addressed before moving forward with the project.	Jan 24th - Feb 14th
Implementation Of Networking And Statistics Portion Of The Project	This section outlines the implementation of client - server portion of this application. After finishing the work in this section, the application should be able to connect client applications to a server located on a remote instance. Once the client application is connected to the system, they should have the ability to submit their created decks, and also view decks provided by other users connected to the application. In addition to the networking portion, the statistics engine should be implemented by this point.	Feb 14th - March 8th
Testing Of Networking and Statistics Engines	During this time, unit tests and integration tests will be implemented within the code base. The unit tests created during this step will ensure that core networking functionality is running as intended, and the integration tests will make sure that the networking communication layer of the application is properly functioning.	March 8th - March 12th
User end testing and bug fixing	At this point, the project should be well in its final stages of development. It is at this time that it should be tested by users within the intended target audience of the application. During this period, users will be asked to use the application and submit feedback on what they thought of the application. In addition, the application will put through a testing cycle, and any software bugs or issues found will attempted to be removed from the final iteration.	March 12th - March 16th
Final Report Writing	With the project code complete, all work will be focused on the final report that is to be submitted alongside the	March 16th - April 12th

	project. During this time, the report for this project will be worked on, completed and submitted by the report deadline.	