# CS400 Project One Proposal

**GROUP: KD FLIPGRID:https://flipgrid.com/spr21kd**

**TA: Keren TA EMAIL: kchen359@wisc.edu**

**RED TEAM (Role: Name - Email):**

**Data Wrangler:** Brenna Buck - 1. bbuck2@wisc.edu

**Backend Developer:** Minyi Li - 2. mli675@wisc.edu

**Frontend Developer:** Muhammad Ismail - 3. mismail6@wisc.edu

**Integration Manager:** Almann Brague - 4. brague@wisc.edu

**BLUE TEAM (Role: Name - Email):**

**Data Wrangler:** Alan Jordao Cortez - 1. ajcortez@wisc.edu

**Backend Developer:** Jack Gundrum - 2. jpgundrum@wisc.edu

**Frontend Developer:** Ryan Stevenson - 3. rstevenson5@wisc.edu

**Integration Manager:** Zhi Zheng - 4. zzheng94@wisc.edu

**Project Title:** Movie Mapper Database.

**Brief Project Description:**

*This application will allow users to explore a movie data set, and find new movies to watch. The data set contains short descriptions, user rating averages, one or more genres, and additional metadata (including title, actors, and director) for each of the movies. Users will be able to select search criteria which consist of one or more genres to select movies by and a range of ratings. Users will be able to select any set of genres, and rating ranges. The system will then display the number of movies that fit the search criteria and the top three of these movies (by rating) with additional details. Users will be able to scroll through the entire list of movies or adapt the search criteria at any time with the app. Primary users of the app will be people who want to find new and interesting movies to watch.*

<brief project description here:

1) what will the app that you are developing do

* provide ratings and reviews for movies depending on user interaction

2) who would use an application like this and why

-those interested in movies because it will give them a detailed overview of each movie

**Data Wranglers:** Brenna Buck Alan Jordao Cortez

The Data Wrangler implements code to read data from a movie database stored in a file, make sure that the movies and their properties are represented accurately in memory, and that errors when reading a movie data file are handled using appropriate exceptions and meaningful error messages.

**Application Data:**

The data will be contained in a text file in CSV (comma separated value) format. It organizes the data into lines: one per movie, with each property of a movie separated by commas. The name of the properties are specified by the first line of the file. The following example shows the first three columns and lines (including the header line) of the data file:

title,original\_title,year,genre,…

La battaglia di Long Tan,Danger Close: The Battle of Long Tan,2019,"Action, Drama, War”,…

Era mio figlio,The Last Full Measure,2019,"Drama, War”,…

The Dirt,The Dirt,2019,"Biography, Comedy, Drama”,…

**Data Format:**

<define the Java type(s) that this data will be stored in after it is loaded>

The movie title, genre(s), name of the director, and the movie description will be stored as a String, the production year will be stored as an Integer, and the average rating for the movie will be stored as a float.

The four types referenced by the interface are type *java.util.List*, *java.io.File*, *java.io.FileNotFoundException*, *java.io.IOException*, and *java.util.zip.DataFormatException*.

**Data Wrangler Deliverables and Deadline:**

<describe how and when this code will be shared with other team roles, note this should be before the group project deadline>

The Data Wrangler will implement these interfaces and test the implementations until Fri, Feb 26

Data Wranglers also need to write and submit at least 2 tests for your implementation with the proposal. (You will need at least 5 tests for your role)

**Back End Developer:** Minyi Li Jack Gundrum

The Back End Developer’s code will make use of the Data Wrangler’s code through the MovieInterface and the MovieDataReaderInterface. While waiting for this code to be written, we will create dummy objects to develop against.

**Data Processing:**

<brief description of the processing that your program will perform on this data, and how it will make use of the hash table data structure>

The class implemented by us will instantiate the Data Wrangler’s implementation of the *MovieReaderInterface* interface in its constructor. It will then use this instantiation to read in the data file passed as the command line argument to the application.

Internally, the class will store lists of movies in a hash table. The hash table will map strings that represent genres and ratings to lists of movie objects of the respective genre or rating.

**Front End Interface:**

<define the Java interface by listing the specific method signatures that exposes this functionality to the front end of your application>

The front end interface will include functions to allow users to retrieve data about movies

Some functions would include addGenre(genre), addAvgRating(rating), getThreeMovies(index), searchByRating(RatingMin, RatingMax)

**Back End Developer Deliverables and Deadline:**

<describe how and when this code be shared with other team roles, note this should be before the group project deadline>

Try to have individual backend tests done by Tues, Feb 16 to get feedback on your methods right after writing them. If either of us have issues we can’t solve ourselves, we can communicate via text, or contact our TA.

Our code will be made available to all other roles by Monday night February 22nd. This will give the other team members enough time to test our actual implementation, rather than just dummy test objects. The front end developer’s need to have their interface working by Fri, Feb 26 so giving them 5 days to make sure everything is working should be an adequate amount of time. Plus this deadline in mind gives us a time cushion just in case it takes longer than expected.

**Front End Developer:** Muhammad Ismail Ryan Stevenson

<brief description of the Front End Developer’s role and responsibilities in this project>

The Front End Developer’s code will make use of the Back End Developer’s code through the BackendInterface. While waiting for this code to be written, they will create a dummy object to develop against.

**Integration with Back End:**

<brief description of how front end and back end will work together and how the front end will instantiate the back end>

The main class developed by the Front End Developer will include the project’s main method through which the Movie Mapper program can be started. The main method takes a file path to a CSV file. The front end is responsible for instantiating, initializing and running the back end through the BackendInterface.

By default, none of the genres and none of the average ratings are selected in the back end when it is instantiated. When the system starts, all ratings will be selected. The front end is responsible for adding all ratings to the back end for this.

**User Commands:**

<brief description of the different modes or screens of the front end and the primary commands that users of your application will invoke>

The front end will have three modes: a base mode, a genre selection mode, and a ratings selection mode.

Every one of the three modes is followed by a command line that lets the user type in numbers and letters.

In the base mode, users will be able to type in numbers to navigate through the list of selected movies. The character ‘g’ will be the command to enter the genre selection mode, character ‘r’ will get the user to the rating selection mode. When the user types ‘x’, the program will exit.

In the genre selection mode, any number from the list of genres will select or deselect the respective genre. The command ‘x’ will cause the program to return to the base mode.

In the ratings selection mode, numbers will select / deselect ratings and ‘x’ will return the user to the base mode.

**Front End Developer Deliverables and Deadline:**

<describe how and when this code will be shared with other team roles, note this should be before the group project deadline>

The Front End Developer will implement this interface and test the implementation until Fri, Feb 26. The implementation will be ready early enough so that other team members will have sufficient time to test their implementations against it in addition to running their code against their own dummy implementation of the front end.

**Integration Manager:** Almann Brague Zhi Zheng

<brief description of the Integration Manager’s role and responsibilities in this project>

The Integration Manager will help other team members clear integration questions and make sure that the development stays on track. They will be responsible for the final integration of the three parts into one application. They will also create a short video demonstrating their team's implementation.

Specifically, the Integration Manager will familiarize themselves with all the interfaces and requirements described in the proposal. They will assist the other team members with implementing their dummy classes and will help discuss and resolve any integration questions. At the same time, the project manager will be responsible for scheduling team meetings during the implementation phase and regularly check in with team members to coordinate the development and integration of all the components. Finally, the Integration Manager will work with their teammates to assemble all the parts of the application and write a Makefile for the project that compiles, runs, and tests the system. Once the system is integrated and fully functional, they will prepare and submit a demo video for the submission and upload it to FlipGrid.

**Integration Milestones (team blue):**

<list brief summaries for the team’s internal deadlines through the final week of this project’s development>

Proposal: 2/16/2021

Kickoff:

Tasks:

1. The integration manager is available for his teammates to ask any early questions about integration and development of the dummy classes.
2. If some team members feel their job is heavier than others, the team can rearrange some contents in the project to make sure each member does contribute.

Due Date 2/19/2021 Through emails or snapchat

Milestone 1

Tasks:

1. Developers and [Data Wranglers](https://canvas.wisc.edu/courses/244796/pages/movie-mapper-data-wranglers) work on their dummy classes and initial versions of implementations.
2. Prepare the interface or class for the next team member to implement. For example, the Data Wrangler will implement the interface to enable the Back End Engineer to load a data file.
3. The integration manager will familiarize himself with all the interfaces and classes in the system.
4. Clear any question that developers have after a few days of working on their implementations.
5. Explain how each part of the system works and talk about how to integrate each part.
6. Solving potential integration problems.

Team meeting before 2/21/2021 Due Date 2/21/2021

Milestone 2

Tasks:

1. All three parts of the system will be ready.
2. Developers and Data Wranglers try to run each part individually in the meeting.
3. The Integration Manager begins to integrate.
4. The remaining integration questions will be discussed.

Team meeting before 2/26/2021 Due Date: 2/26/2021

Milestone 3

Tasks:

1. The system will be integrated and the Makefile will be ready to compile, run, and test the system.
2. The integration manager shows the team the process to run the whole system.
3. Fix last integration bugs.
4. Help the integration manager decide on the contents of the demonstration video.
5. Then the integration manager will finish the demonstration video.

Team meeting before 3/1/2021 Due Date:3/1/2021

**Integration Milestones (team red):**

<list brief summaries for the team’s internal deadlines through the final week of this project’s development>

Proposal: 2/16/2021

Kickoff:

Tasks:

1. Every member of the team familiarize themselves with their role and responsibilities for the project
2. If there are questions or concerns about their portion of the project the integration manager is available to help with any questions.

Due Date: 2/19/2021

Milestone 1:

Tasks:

1. The front end developer, back end developer, and data wrangler will work on their dummy class as well as the initial integration.
2. Each team member will their interface or class so that each team member can smoothly implement each portion of the project
3. The integration manager during this time will familiarize themselves with each interface that individual members are working on. As well as be available to answer any questions that arise throughout the process of creating these interfaces.
4. A final meeting will be held where each interface will be explained and a plan for overall integration will be created. In this time any initial problems with integration will be discussed.

Due Date: 2/22/2021

Milestone 2:

Tasks:

1. Each member will have completed their section of the project.
2. Each member will run their individual portion of the project so the entire group can see their final interface as well as ask any questions that may arise.
3. The integration manager will start implementing the three sections of the project.
4. Any overall questions will be discussed

Due Date: 2/26/2021

Milestone 3:

Tasks:

1. The final integration will be completed so the Makefile will be able to compiled, run, and tested. In this the integration manager will run through the entire process of the system.
2. If problems arise during the first portion they will be able to be fixed during this meeting.
3. The final demonstration video will be planned out during the meeting and then the integration manager will complete the final demonstration video.

Due Date: 3/01/2021

**Demonstration Video:**

<describe how you plan to demonstrate the capabilities of your application in the final demo video for this project>

For the final demo video we will have the integration manager run through how the final application works. We will do this through screen sharing and a live camera of the integration manager who will be explaining each step of the process. To start we will introduce the initial home screen of the application and run through the options the user has. Then from there we will navigate through the program showing off the different functions of the application. Some of these functions will include showing how users can search through the entire library of movies individually. As well as showing how users can narrow their search results based on the specific movie genres they prefer or to narrow it based on the ratings that the movies have received. Once this recording is complete we will upload the final video to FlipGrid.

**Additional Responsibilities and Notes:**

<list by role, any additional responsibilities that are expected of team members to help balance the workload for this project… if you are concerned about your project being too simple or too involved, this is also a good place to suggest plans for expanding or contracting your main idea>

* Generally for all roles try to meet their given deadlines. If you are having difficulties ask the person who has the same role as you, or ask our TA, Keren, to see if he can help.
* Every role should stick to what it says on their giving canvas page on what to do. Doing so will allow the workflow to be evened out. The instructors assign these roles for a purpose, and we know that we will be given different roles each project. Therefore, if we each do what it says on our role’s page, for each project, it is likely that all will have done roughly the same amount of work(specifically in the different projects where it seems like role A did way less work than role B, switching roles evens it out). Help from other team members is always appreciated though. For example, If someone is the Integration Manager role they can suggest their own helpful tactics to other roles should they need it.

**Team Signatures:**

Each team member must type their own name and current date in place of one of the blanks below, to confirm their contributions and agreement to the proposal described above.

Jack Gundrum 02/15/21

Brenna Buck 02/15/21

Ryan Stevenson 02/15/21

Muhammad Ismail 02/15/21

Almann Brague 02/15/21

Zhi Zheng 02/15/21

Minyi Li 02/16/21

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**End of Proposal**