**Part 1 - Work Distribution**

**What each member has been working on and future plans**

Bryan Guo

Worked on GameManager and Board classes, created a builder design pattern for the construction of the GameManager and its associated Board.

Wrote unit tests for Board, GameManager, and BoardManager.

Future plans: Develop more thorough test cases, help polish controllers and other use cases, integrate design patterns and smoother code relations.

Muhammad Ibrahim

Developed Board and BoardManager classes, which incorporated all of the logic involved with checking whether legal chess moves can be made. Also worked on all of the other entity classes.

Future plans: fix bugs across the BoardManager method and eliminate code smells in existing use case classes. Do whatever work is necessary to create a finished, functioning chess program.

Vala Jalalvandi

Worked on the controller classes, with most focus on the GameActivity class

Redesigning the layout of the board to make it dynamic with the Board class

Adding the ability to move the pieces by clicking on them on the board.

Juan Martin

First steps of the app: basic GUIs, Activities…

Integration of the Firebase API into our project.

Create the matchmaking logic.

Future plans: continue working on the game state and implement SQLite database to save common user data.

Jaren Worme

Helped group members with different aspects of the code, namely the make move logic across the controller and entity classes.

Future plans: Eliminate any code smells, work on a new clock method to time moves to increase the scope of the project.

**Part 2 - Things That Have Worked Well With Our Design**

Bryan Guo

The use of a GameBuilder will likely be useful for additional features that may be added in Phase 2. Overall, the separation of layers in our architecture is clear, which allowed us to code independently and push changes to github.

Muhammad Ibrahim

We were able to effectively use Git and Githubs in creating our software and making progress across the different layers of architecture. In addition, we were able to successfully apply OOP concepts such as abstraction, inheritance, composition and polymorphism to ensure that our work met the criteria.

Vala Jalalvandi

The separation of the different layers of architecture has allowed us to work together more effectively, and it serves a reminder to adhere to the clean architecture design as much as possible

Juan Martin

Thanks to our design we are able to split the work easily and everyone can work independently and merge their versions without too many conflicts.

Jaren Worme

Separating Our classes and sorting by packages made everything easy to understand and work with.

**Part 3 - Current Struggles And Open Questions**

Bryan Guo

Specific moves in the chess game are difficult to check, example being castling and en passant. Specifically, the implementation of en passant may require a unique implementation method, or even the use of a design pattern.

Muhammad Ibrahim

Figuring out a way to eliminate code smells in the BoardManager class whilst ensuring that my code is functional with the other components of the software. Also, we need to figure out a way to allow castling to occur, alongside pawn promotions.

Vala Jalalvandi

There are quite a few bugs that still need to be figured out in the code. These bugs sometimes cause the program to crash unexpectedly, and they need to be cleared out. There are also some concerns about whether the current design we have implemented is scalable to more features that will be added in the future.

Juan Martin

We think our design is good but we are not sure if it is the most appropriate for the project.

After having basic chess games working, where should we go next?

Jaren Worme

Since my role was more to help across the project, I did not have a specific code section that I alone was working on. I can rectify this in phase two with a clock class I intend to implement. Also, due to the nature of our program, it has been difficult to implement many design patterns apart from builder, which we will look into for phase 2.