Project Schedule:

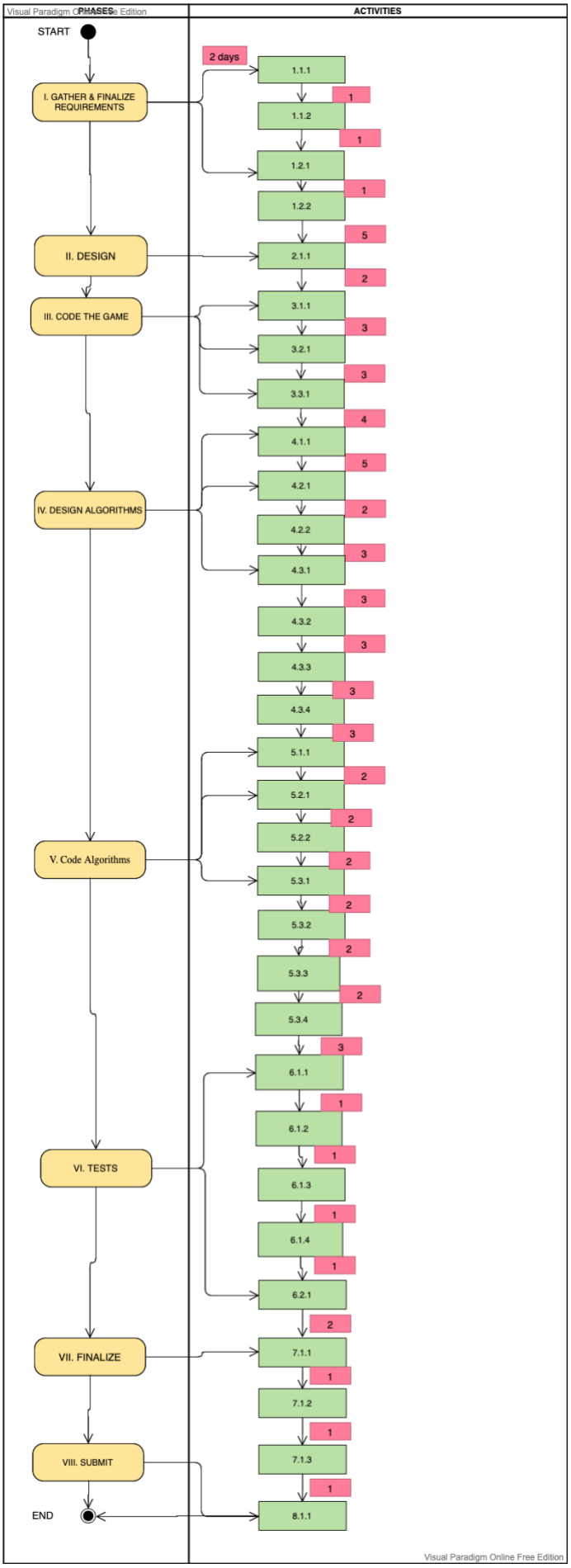
PHASES	STEPS	ACTIVITIES
I. GATHER & FINALIZE REQUIREMENTS	1. Gather Requirements	1.1.1. Understand what requirements are needed to build the chess game. 1.1.2 Jot down all the requirements needed to build the chess game.
	2. Finalize Requirements	1.2.1. Filter through all the requirements and take out the unnecessary ones. 1.2.2. Document all the final requirements.
II. DESIGN	1. Create System Design Model Structure	2.1.1. Understand what is needed in the System Design Model Structure and build it out.
III. CODE THE GAME	1. Chess boards	3.1.1. Code the chess board
	2. Players	3.2.1. Code the chess players
	3. Users	3.3.1. Code the users who will want to play
IV. DESIGN ALGORITHMS	1. Openings	4.1.1. Design the algorithm for different chess openings
	2. Movement	4.2.1. Design the algorithm for different movements 4.2.2. Design the algorithm for when players cannot make illegal moves
	3. Closing	4.3.1. Design the algorithm for when a player wins the game 4.3.2. Design the algorithm for when it is a draw 4.3.3. Design the algorithm for when it is a draw 4.3.4. Design the algorithm for insufficient material
V. CODE ALGORITHMS	1. Openings	5.1.1. Code the algorithm for different chess openings
	2. Movement	5.2.1. Code the algorithm for different movements 5.2.2. Code the algorithm for when players cannot make illegal moves
	3. Closing	5.3.1. Code the algorithm for when a player wins the game 5.3.2. Code the algorithm for when it is a draw

		5.3.3. Code the algorithm for when it is a draw 5.3.4. Code the algorithm for insufficient material
VI. TESTS	1. Run Tests	6.1.1. Run tests on the chess board 6.1.2. Run tests on the players 6.1.3. Run tests on the users 6.1.4. Run tests on the algorithms
	2. Debugging	6.2.1. Debug any errors in the code from the test runs
VII. FINALIZE	1. Finalize all code	7.1.1. Make sure all code passes all the tests. 7.1.2 Clean up any left-over bugs. 7.1.3. Finalize all code.
VIII. SUBMIT	1. Submit	8.1.1. Submit the completed project.

Activity Graph:

(See next page)

The estimated time cost of each activity is in the pink boxes listed in days. This project will probably take about 66 days (approximately 2 months) to complete from start to finish. Although, it could be less. The time available to complete the project is 3 months (93 days; December). The estimated time to complete the project is 66 days. The slack time is 27 days (about 1 month). The critical path is 27 days (about a month). This project can be finished before May.

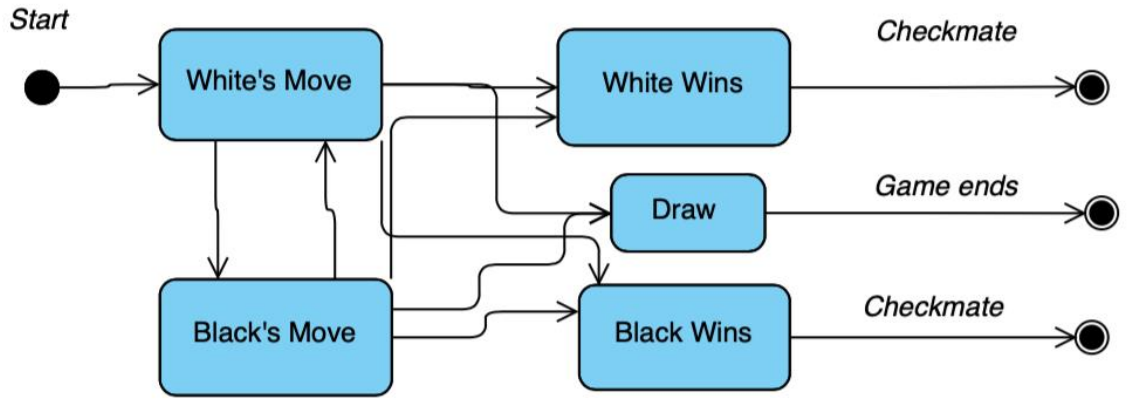

Visual Paradigm Online Free Edition
PHASES
ACTIVITIES
START
I. GATHER & FINALIZE REQUIREMENTS
II. DESIGN
III. CODE THE GAME
IV. DESIGN ALGORITHMS
V. Code Algorithms
VI. TESTS
VII. FINALIZE
VIII. SUBMIT
END
2 days
1.1.1
1
1.1.2
1
1.2.1
1
1.2.2
5
2.1.1
2
3.1.1
3
3.2.1
3
3.3.1
4
4.1.1
5
4.2.1
2
4.2.2
3
4.3.1
3
4.3.2
3
4.3.3
3
4.3.4
3
5.1.1
2
5.2.1
2
5.2.2
2
5.3.1
2
5.3.2
2
5.3.3
2
5.3.4
3
6.1.1
1
6.1.2
1
6.1.3
1
6.1.4
1
6.2.1
2
7.1.1
1
7.1.2
1
7.1.3
1
8.1.1
Visual Paradigm Online Free Edition

Functional & Non-Functional Requirements:

Some of the functional requirements for the chess game are the way in which the pieces move and the way in which the players are captured, etc. This will also include what moves are legal, promotion (getting promoted from one piece to another), the way the game saves or loads, network connectivity, and message prompts for different phases of the game (i.e., if the game ends, a message prompt that says “Checkmate” pops up).

Some of the non-functional requirements for the chess game are that the chess game should not crash whilst it is being used. The chess game should also be capable of being installed on any computer’s operating system. The chess game should also not pose any security risk to the user’s computer. Another non-functional requirement would be making sure the chess game has enough network bandwidth to handle the game overall.

State Machine Diagram:



In the game of chess, there are alternating moves between black and white. When the game starts, white moves first. The game can draw if white and black keep moving back and forth within the same spots. The game can also end when it is either white or black’s turn. The game can also end by a moving player winning, which would be checkmate, or resigning.