Week-1

**Page 1: Interview Report**

**Part 1: Requirements Gathering via an Interview**

**Mock Interview Simulation**

**Interviewer: Student Software Developer**

**Interviewee: Battleship Game Expert (Mock Role)**

Q1: Can you describe how the traditional Battleship game is played?

A: Battleship is a turn-based strategy game where two players arrange a fleet of ships on a hidden grid and take turns guessing the locations of the opponent’s ships by calling out grid coordinates. The goal is to sink all of the opponent's ships.

Q2: What are the key elements of the game that must be preserved in a software version?

A: Essential elements include two 10x10 grids (one per player), different types of ships (with varying lengths), turn-based play, feedback on hits and misses, and a way to track sunk ships. It should also preserve the suspense and strategy of locating hidden ships.

Q3: How are the ships placed in the game?

A: Ships are placed either vertically or horizontally, and they must not overlap. Each player places their ships on their own grid without the opponent seeing.

Q4: What types of players will use this software?

A: Casual gamers, students learning logic or strategy, and potentially AI or solo players for training. It should support both two-player and single-player modes.

Q5: What are the main challenges in implementing Battleship as a digital game?

A: Ensuring intuitive UI/UX for grid input and ship placement, enforcing game rules programmatically, handling turn-based logic, and providing clear feedback on hits, misses, and game progress.

Q6: What optional features could enhance gameplay in a digital version?

A: Features like AI difficulty levels, sound effects, visual animations for hits/sinks, a replay feature, and online multiplayer support would add value.

Page 2: User Stories (1–4)

User Story 1

As a red player,

I want a grid to track my ship placements,

So that I can visualize my fleet and manage my defense.

Acceptance Criteria:

A 10x10 grid is displayed to the user.

Player can place five ships of various lengths.

Ships cannot overlap or be placed diagonally.

User Story 2

As a red player,

I want to select grid coordinates for my attacks,

So that I can try to sink the blue player's ships.

Acceptance Criteria:

The player selects a coordinate from the opponent’s grid.

The system confirms whether it was a hit or miss.

The result is marked on the player's tracking grid.

User Story 3

As a red player,

I want feedback after each turn,

So that I can know if I’ve hit, missed, or sunk an enemy ship.

Acceptance Criteria:

After selecting a target coordinate, a message appears with the result.

If a ship is completely hit, the system announces it has been sunk.

User Story 4

As a new player,

I want a tutorial mode,

So that I can learn how to play the game step-by-step.

Acceptance Criteria:

The tutorial introduces game objectives and rules.

Interactive prompts guide the player through ship placement and turn-taking.

Feedback and explanations are provided for actions taken.

Page 3: User Stories (5–7)

User Story 5

As a solo player,

I want to play against a computer opponent,

So that I can play when no other person is available.

Acceptance Criteria:

An AI opponent randomly places ships.

The AI selects attack coordinates following basic logic.

The game alternates turns between the user and AI.

User Story 6

As a player,

I want a way to reset the game,

So that I can start over at any time.

Acceptance Criteria:

A “Restart Game” button is visible on the interface.

Clicking the button clears the current game state.

Players are returned to the setup phase.

User Story 7

As a player,

I want to view a summary of the game when it ends,

So that I can see how the match progressed.

Acceptance Criteria:

A game-over screen displays winner and number of turns taken.

The summary includes statistics like hits, misses, and sunk ships.

An option to play again is presented.