­­

**Battleship application**

**Table of Contents**

[1 Introduction 3](#_Toc491271204)

[2 Technologies and requirements 3](#_Toc491271205)

[3 Overview 3](#_Toc491271206)

[In scope (must-have) 3](#_Toc491271207)

[4 Workflow 4](#_Toc491271208)

[UI Design Considerations 4](#_Toc491271209)

[General Constraints 5](#_Toc491271210)

# Introduction

This application is used to test frontend candidates. Please keep in mind that this is the project that will be used to evaluate your skills. The project will be evaluated as if you are delivering it to a customer. We expect you to make sure that the app is fully functional and doesn’t have any obvious missing pieces. Please ensure that you’ve read and understood all requirements.

Please create a repository on GitHub and share the link with us once you’re done. We’d like to be able to see your thought process, so we’d appreciate smaller commits where it makes sense.

# Technologies and requirements

* Frontend: JavaScript or TypeScript, React. You may feel free to use Redux, LESS, or modern ES6 JavaScript features if you’d like.
* Backend: no

# Overview

## In scope (must-have)

The application is a partial Battleship game as a web app. The computer has positioned five ships of various sizes on a 10x10 board. Each ship must be placed horizontally or vertically, completely on the board, without overlapping another ship. The player cannot see the ship locations. Each round, the player “fires” at a board position of their choice. The computer indicates if this was a “hit” or a “miss”. When all tiles of a particular ship have been hit, the computer indicates that the entire ship has been sunk. When the player has sunk all the ships, the game is over. Obviously, this game would be more fun if the player had their own ships and the computer was firing back, but we’ll leave that out for simplicity. In other words, we are only implementing the turns for Player 1, not for Player 2. You may use the provided JSON data (see below) indicating the position of the ships. You should produce a web app for this game as described, according to the provided mocks. The game should be responsive and mobile-friendly, so that it may be played on an iPhone 5-sized screen (320x568) up to a desktop browser (approx. 1440x1024). It’s not necessary to save game state or anything like that.

Ship layout data:

{

"shipTypes": {

"carrier": { "size": 5, "count": 1 },

"battleship": { "size": 4, "count": 1 },

"cruiser": { "size": 3, "count": 1 },

"submarine": { "size": 3, "count": 1 },

"destroyer": { "size": 2, "count": 1 }

},

"layout": [

{ "ship": "carrier", "positions": [[2,9], [3,9], [4,9], [5,9], [6,9]] },

{ "ship": "battleship", "positions": [[5,2], [5,3], [5,4], [5,5]] },

{ "ship": "cruiser", "positions": [[8,1], [8,2], [8,3]] },

{ "ship": "submarine", "positions": [[3,0], [3,1], [3,2]] },

{ "ship": "destroyer", "positions": [[0,0], [1,0]] }

]

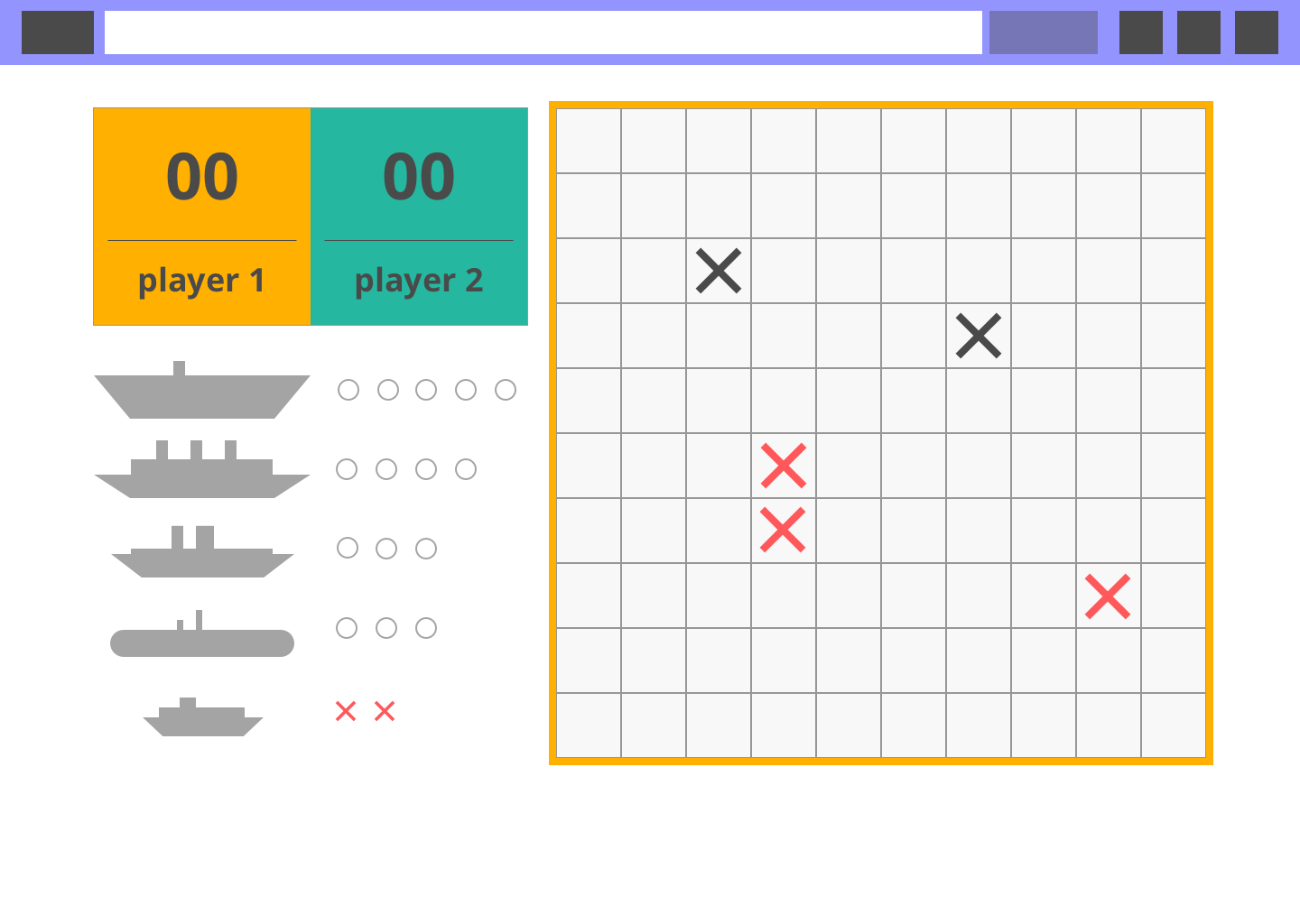
}

# Workflow

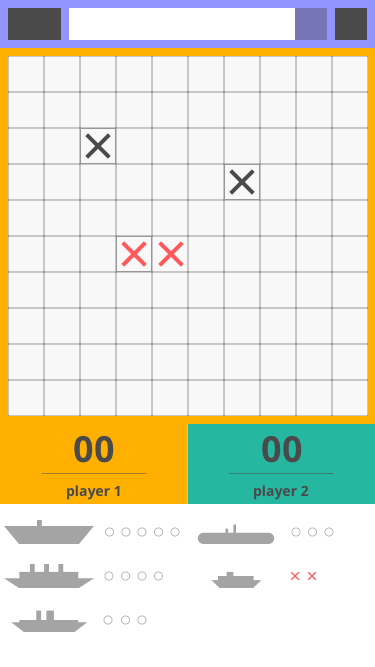
## UI Design Considerations

Please use assets provided in the archive “assets”. Find sample screens for different resolutions below.

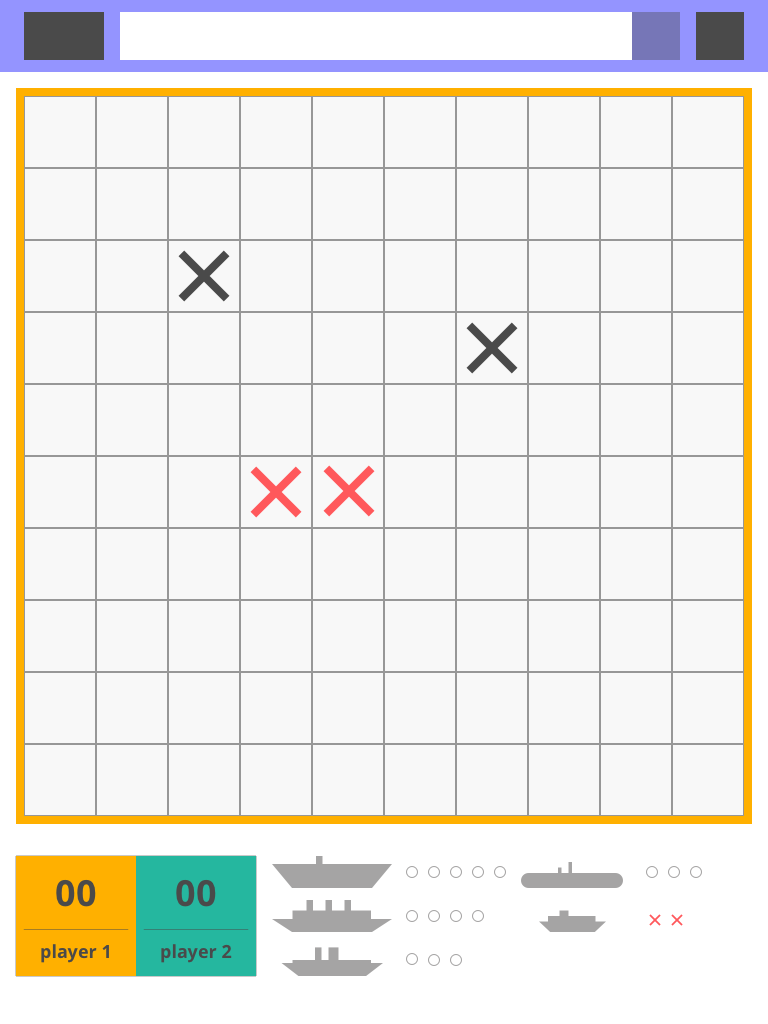
Desktop:



Mobile:



Tablet:



## General Constraints

**No plugin** (flash, silverlight) should be used for development.