**Project Milestone 4**

---

*CSCI 3308: Software Development & Tools*

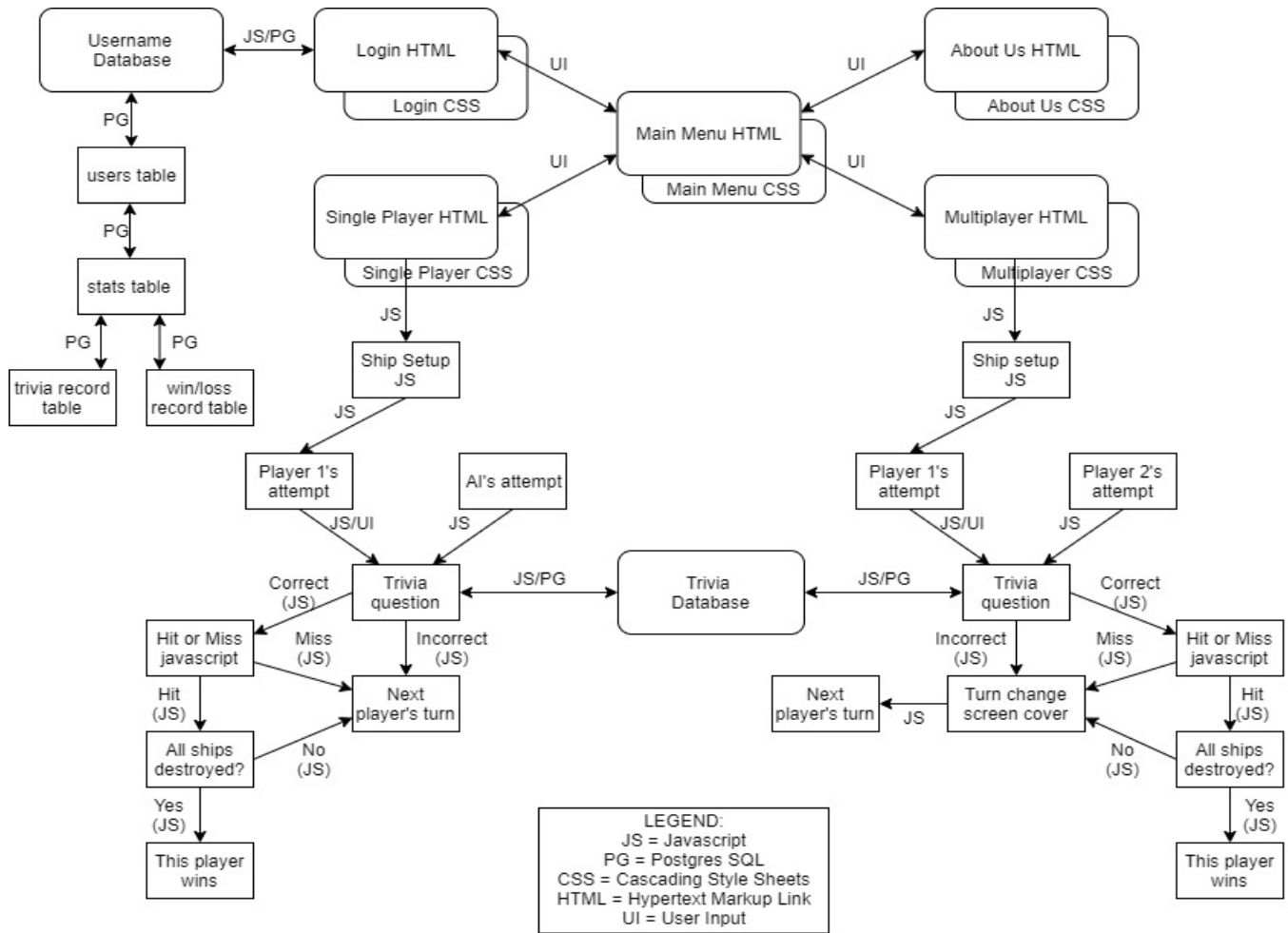*Team 113-5: History Through Battleship*

Team Members
- Salvatore Pacifico
- David Crockett
- Matthew Ruiz Diaz
- Andres Montealegre
- Kai Schuyler Gonzalez

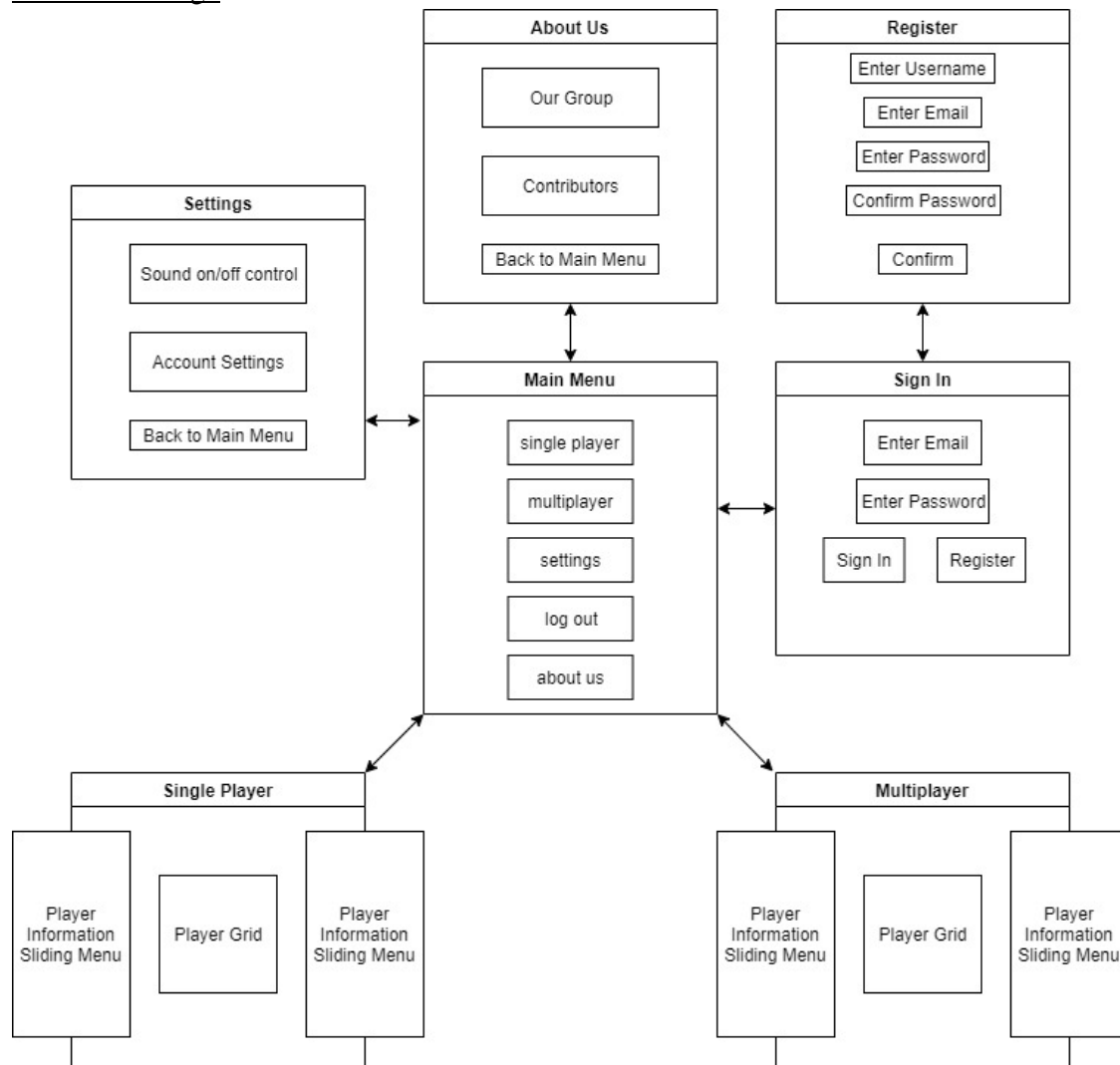Revised List of Features (Ranked by Priority)
1. Single and Multiplayer modes - Two modes, where single player is against an AI and local multiplayer is against someone playing on the same system
2. Trivia Minigame - A popup multiple choice question that the user has to get right in order to hit the opponent's battleship. If the user gets the question wrong, they will not know if they hit a ship or not.
3. Trivia Question API - using Open Trivia DB API to get the questions for the Trivia Minigame.
4. Main menu - Have a screen where users can select between local or multiplayer game modes, settings, about us section, and a win/loss record section
5. Login System - Have users login with a password and username where their game stats will be saved, option to change password or username
6. Settings Menu- A menu that has account options and sound options
7. End Game Screen  - Having a "you win" or "you lose" screen at the end of each game with the option to play again or quit to the main menu
8. Alternating Screen - Having a screen that shows up during the local multiplayer mode that prevents users from seeing each other's placement of ships.
9. Sound - Sound effects will be included for a hit/miss and for right/wrong answers.
10. Win/Loss Stats - A menu option where one can view their single/local multiplayer stats. The stats will be stored in our Postgres database.
    ***Most features did not change, only priority***

## Architecture Diagram



**Username Database** →(JS/PG)→ **Login HTML** / Login CSS →(UI)→ **Main Menu HTML** / Main Menu CSS →(UI)→ **About Us HTML** / About Us CSS

Username Database ↕(PG) **users table** ↕(PG) **stats table** ↕(PG) **trivia record table**, ↕(PG) **win/loss record table**

**Main Menu HTML** ↔(UI) **Single Player HTML** / Single Player CSS

**Main Menu HTML** ↔(UI) **Multiplayer HTML** / Multiplayer CSS

Single Player CSS →(JS)→ **Ship Setup JS** →(JS)→ **Player 1's attempt**

Multiplayer CSS →(JS)→ **Ship setup JS** →(JS)→ **Player 1's attempt**, **Player 2's attempt**

**Player 1's attempt** →(JS/UI)→ **Trivia question**
**AI's attempt** →(JS)→ **Trivia question**

**Trivia question** ↔(JS/PG) **Trivia Database** ↔(JS/PG) **Trivia question**

**Player 1's attempt** →(JS/UI)→ **Trivia question** (right side)
**Player 2's attempt** →(JS)→ **Trivia question** (right side)

Left side:
Trivia question —Correct (JS)→ **Hit or Miss javascript**
Trivia question —Miss (JS)→ **Next player's turn**
Trivia question —Incorrect (JS)→ **Next player's turn**
Hit or Miss javascript —Hit (JS)→ **All ships destroyed?**
All ships destroyed? —No (JS)→ **Next player's turn**
All ships destroyed? —Yes (JS)→ **This player wins**

Right side:
Trivia question —Correct (JS)→ **Hit or Miss javascript**
Trivia question —Miss (JS)→ **Turn change screen cover**
Trivia question —Incorrect (JS)→ **Turn change screen cover**
Turn change screen cover —JS→ **Next player's turn**
Hit or Miss javascript —Hit (JS)→ **All ships destroyed?**
All ships destroyed? —No (JS)→ **Turn change screen cover**
All ships destroyed? —Yes (JS)→ **This player wins**

**LEGEND:**
JS = Javascript
PG = Postgres SQL
CSS = Cascading Style Sheets
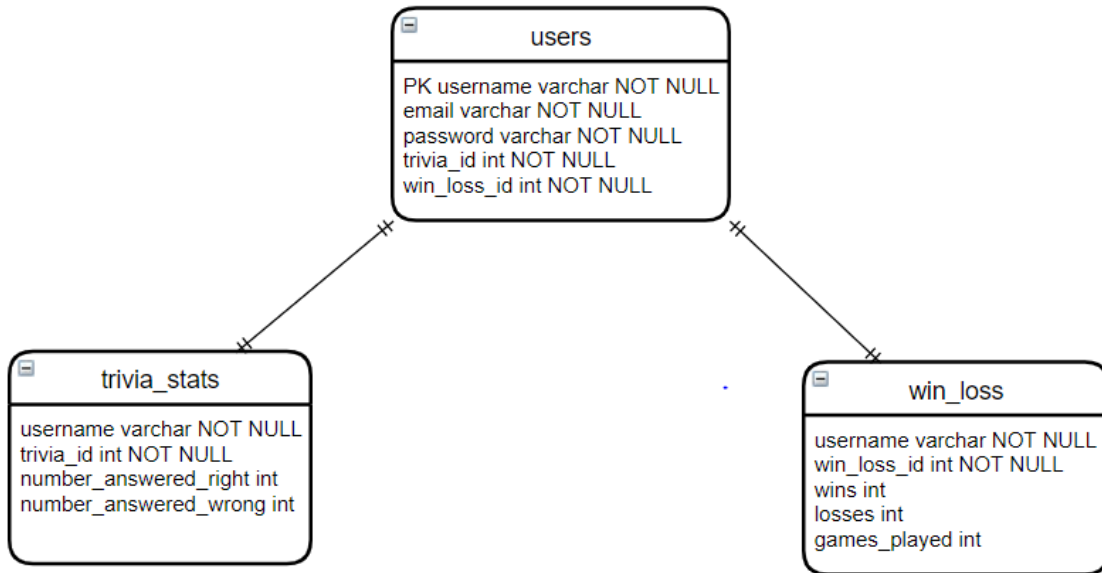HTML = Hypertext Markup Link
UI = User Input

Front End Design



Web Service Design
- We are using an API called Open Trivia DB for the trivia implementation in our game. The API will be called in a javascript onclick() function and we will be using session tokens to ensure users do not get the same question multiple times. The API documentation can be found here.

Database Design

**users**

PK username varchar NOT NULL
email varchar NOT NULL
password varchar NOT NULL
trivia_id int NOT NULL
win_loss_id int NOT NULL

**trivia_stats**

username varchar NOT NULL
trivia_id int NOT NULL
number_answered_right int
number_answered_wrong int

**win_loss**

username varchar NOT NULL
win_loss_id int NOT NULL
wins int
losses int
games_played int

---

Individual Contributions

**Andres: Latest commit**
- Battleship Button Grid
- Button Disable Feature (Post-Correct Answer)
- NavBar Responsive Implementation

**Sal: Latest commit**
- Main Menu HTML/CSS
- About Us CSS
- Login/Register CSS
- Connected HTML Pages
- Database Diagram
- Settings HTML/CSS
- Side Nav Design
- Ship Placement Functionality
- Start Game Validation

**Kai: Latest commit**
- Made the CSS responsive and merged our CSS files to create a standard CSS file to use in all pages of the site
- Installed Postgres on the Heroku app
- Created Postgres Node.js app
- Login screen HTML

**Matthew: Latest commit**
- trivia popup css
- modal id design
- wireframes for main menu, settings, and login

**David: Latest commit**

- Architecture Diagram
- Front End Design
- About Us HTML



Projects / HistoryThroughBattleship-113-5 / H15 board

## Sprint 3, Local Battleship

⚡ ☆ ⏱ 4 days remaining    Complete sprint    ⌁    ...

10/4-10/31 First two weeks involve planning, coding | Last week involves testing, implementing

🔍    KG AA DC MD SP ⬤    Only My Issues    Recently Updated

| TO DO | IN PROGRESS | DONE |
|---|---|---|
| As a set of users playing locally, I want a screen between moves that will hide what my opponents screen will look like. | As a user, I want to be able to look through my win/loss stats. | As a user, I want there to be a large database of trivia questions so I don't encounter the same question multiple times |
| Gameplay | Interface | Database |
| ▢ ↑ 3    H15-13 AA | ▢ ↑ 1    H15-26 KG | ▢ ↑ 3    H15-16 KG |
| As a user, I want sounds to play when there is a hit/miss and for right/wrong answers. | | As a user, I want the battleship game to implement trivia questions so that students can learn history while playing. |
| Gameplay | | Gameplay |
| ▢ ↑ 1    H15-17 MD | | ▢ ↑ 2    H15-6 SP |
| As a user, I want a end game screen so I know when the game is over and whether or not I won. | | As a user, I want to be able to start a local multiplayer game within the app so that users have the option to play with a friend. |
| Interface | | Gameplay |
| ▢ ↑ 1    H15-18 KG | | ▢ ↑ 3    H15-5 DC |

Challenges/Risks
- AI implementation for Single Player mode
  - No Backup plan (hopefully not needed), as this is one of the main functionalities of our project, just a challenging one. If all else fails, this game will only include a local multiplayer (unlikely).
- Tracking user statistics by username
  - Our backup plan would be to include only the login database if time constraints do not allow us to implement this.
- Sound implementation
  - We have found challenges in chrome's ability to support sound, our plan is to include sounds for hits/misses/etc but if time constraints do not allow us to fully implement that, we will only plan on including background music.