

# Project Evaluation

By: Corbin Robinson, Ethan Smith, Zaniken Gurule, Parker Rosenberg

## **Customer Satisfaction Survey**

We interviewed Bo Stevens, a user from our website.

- 1. Did you find familiar games to add to your profile?**
  - a. Yes.
- 2. Did you like the recommendations it gave?**
  - a. They seemed interesting.
- 3. Would you recommend this website?**
  - a. I would recommend it to people that like board games.
- 4. What would you add to make this a better website?**
  - a. More visual elements.
- 5. On a scale of 1-10, what would you rate the usability of the website?**
  - a. 10
- 6. On a scale of 1-10, what would you rate the aesthetics of the website?**
  - a. 5, the jpegs weren't great.
- 7. On a scale of 1-10, how would you rate the website overall?**
  - a. 8

## Document Reflections

- **Process model justification**
  - It wasn't necessarily useful, but it was fairly accurate. The process of making the project followed the waterfall method and was pretty straight forward.
- **Project Proposal**
  - Helpful for figuring out the viability of our project as a product, helped see who is our primary audience, and how the project is being financed.
- **Software Requirements Documentation**
  - Creating it was helpful in setting a more defined path for the project. It wasn't referenced very many times during the actual process, but by clearly defining our requirements, it made the whole thing easier.
- **Software Design Documentation**
  - Useful for showing a breakdown of exactly what components the project would have, useful for breaking the project into pieces to be worked on separately, and understanding exactly what each portion of the project would do.
- **Testing Documentation P1**
  - Useful for giving us ideas of what needed to be tested, what behaviors were expected, and what should not be expected, and helped us design in a way where we could consider what tests were going to be run, so we could think about making the tests pass before we even did any testing.
- **Testing Documentation P2**
  - Helpful in making us run the tests, letting us find some bugs that we might not have otherwise found, (including 1 critical bug) and were able to correct them.
- **Team Member Evaluations**
  - Both useful and accurate, it was the best way for us to evaluate our group members based on their involvement or lack thereof, and accurate because they were private and individual.
- **Complete Project Evaluation**
  - Accurate as it could be, a good reflection of the project as a whole. Not super useful in the grand scheme of this project because it is finished. It would be useful if we were to ever do another project and we would be able to reflect on how this project went. We could use it to reflect on this development process and hopefully improve the next project and its development process.

## SRS Requirements Review

- Functional Requirements -
  - Give Recommendations -
    - Accepts 1 or more board games as input and recommends 1 or more board games based on some of the attributes involved: mechanics, theme, and difficulty.
      - \*Requirement Met\*
  - User Logins -
    - User can log into an account if they enter the username and password correctly.
      - \*Requirement Met\*
    - Systems gives user access to account matching the username/password given.
      - \*Requirement Met\*
    - Regular user logon/Admin logon.
      - \*Requirement Met\*
  - Login Errors -
    - The system should display an error message when a user enters an incorrect username/password.
      - \*Requirement Met\*
  - User Interaction -
    - Users will make their selections by choosing from a drop down list or a game they like.
      - \*Requirement Met\*
    - Selections involve some of the attributes: mechanics, theme, and difficulty.
      - \*Requirement Met\*
- Nonfunctional Requirements -
  - Board game database contains at least 20 board games.
    - \*Requirement Met\*
  - WebApp works on Google Chrome and FireFox.
    - \*Requirement Met\*

**What we learned (as an educational experience):**

We learned that software engineering is a complex topic and is much deeper and more complex than what you originally think of writing software. There's lots of stuff that's important to consider to keep on track, and while some of it is frustrating in the moment it makes sense why it would be nice to have design and consistency.

**If we were to do it again:**

We would understand better what we needed to build and design before we start actually programming, so we would be able to design our project way better.