Project4

# Project Overview

This project aims to build a website that enables friends and family's to share video games.

We all know how fun it is to play a good video game, but the problem is once you conquer all the required quests or battles you reach the end of the game. Even though the game is now over you still do not want to get rid of the game because it was so much fun to play. Maybe you will go back to playing it in a few months or years. Here lies the dilemma, does somebody else want to use it while you are not?

This is where our website comes into play.

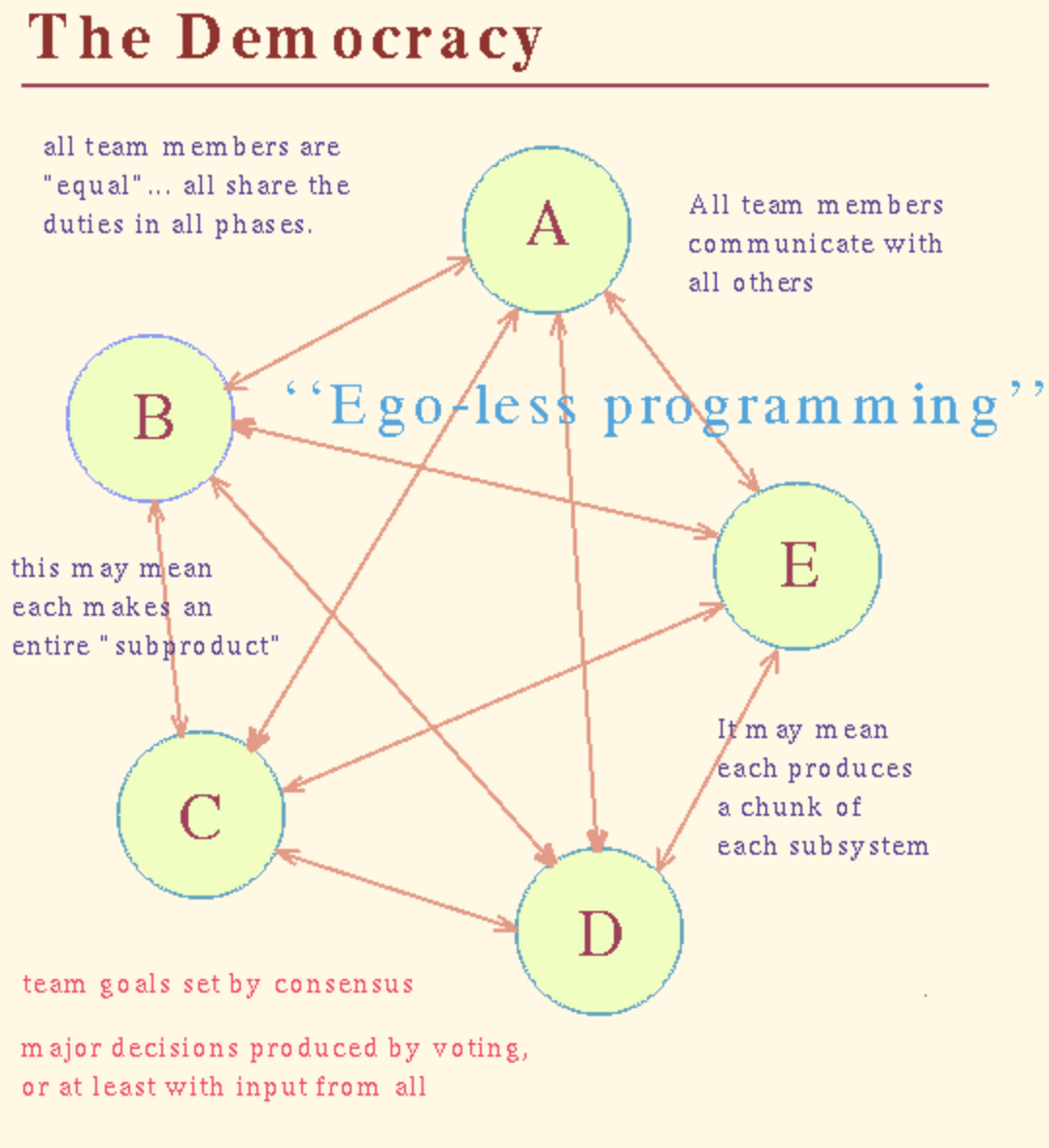
Have you ever been with a group of your friends and wanted to play some sort of game? You take a trip over to the place where the games are stored and realize that all the games have been overplayed? To reduce this problem, It would be nice if there was a way that one could have the ability to look through a library of games, and use that given game without have to run to the store and spend twenty to eighty dollars. Not only are some games rather expensive but, the newly purchased game will soon become overplayed. Later that week you end up talking to your friend that lives two doors down about the awesome game your just purchased to find that he has the exact game in his collect of dust collectors. This is where the game sharing website will came into play.

From the website users would be able to browse the assortment of games per category to find the right game for the night. The categories would enable the users to find the type of game they were looking for, depending if they wanted a specific number of players or wanted to find a game that could be completed quickly. Users could have the ability to upload not there own game to the database or remove games depending if they no longer wanted to loan it out. The typical user would have the ability to send invited to their friends and family via e-mail. When a game has been “checked out” the owner of the game could mark it that way on the website. If the owner would like they could even enable a list that keep track of the next person that would like to check the game out. The website would have features that allow the user to check and out as fast as they could as we all know how precious time is.

As we want this database for this site to continually grow between family and friends we will have a feature the enables uses to request for new or updated features. This way everybody loves the product and it can spread like wild fire.

# Team Organization

The organization that will be used in our program is the formation of the democratic team, this is also known as the egoless team. Below is a diagram representing the organization that will be used.



With the formation of the Democratic Team we will be able to work on the project together. The communication in our team will be transparent and allow everybody to work on task together. The majority of the work will be assigned to individuals, but will be completed as a team effort. This way everybody will be able to work on the task they have the most experience and assist others where needed.

# Software Development Process

The development will be broken up into five phases. Each phase will be a little like a Sprint in an Agile method and a little like an iteration in a Spiral process. Specifically, each phase will be like a Sprint, in that work to be done will be organized into small tasks, placed into a “backlog”, and prioritized. Then, using on time-box scheduling, the team will decide which tasks the phase (Sprint) will address. The team will use a Scrum Board to keep track of tasks in the backlog, those that will be part of the current Sprint, those in progress, and those that are done.

Each phase will also be a little like an iteration in a Spiral process, in that each phase will include some risk analysis and that any development activity (requirements capture, analysis, design, implementation, etc.) can be done during any phase. Early phases will focus on understanding (requirements capture and analysis) and subsequent phases will focus on design and implementation. Each phase will include a retrospective.

|  |  |
| --- | --- |
| **Phase** | **Iteration** |
| 1. | Phase 1 - Requirements Capture |
| 2. | Phase 2 - Analysis |
| 3 | Phase 3 - Architectural, UI, and DB Design |
| 4 | Phase 4 - Detailed Design, Implementation, and Unit Testing |
| 5 | Phase 5 - More Implementation and Testing |

We will use Unified Modeling Language (UML) to document user goals, structural concepts, component interactions, and behaviors.

# Communication policies, procedures, and tools

The main communication method

For general discussion we have a discord channel setup. Additional rooms can be made in order to have team members a way to have isolated conversations as needed if some team members are not needing to see constant chat messages.

Procedures

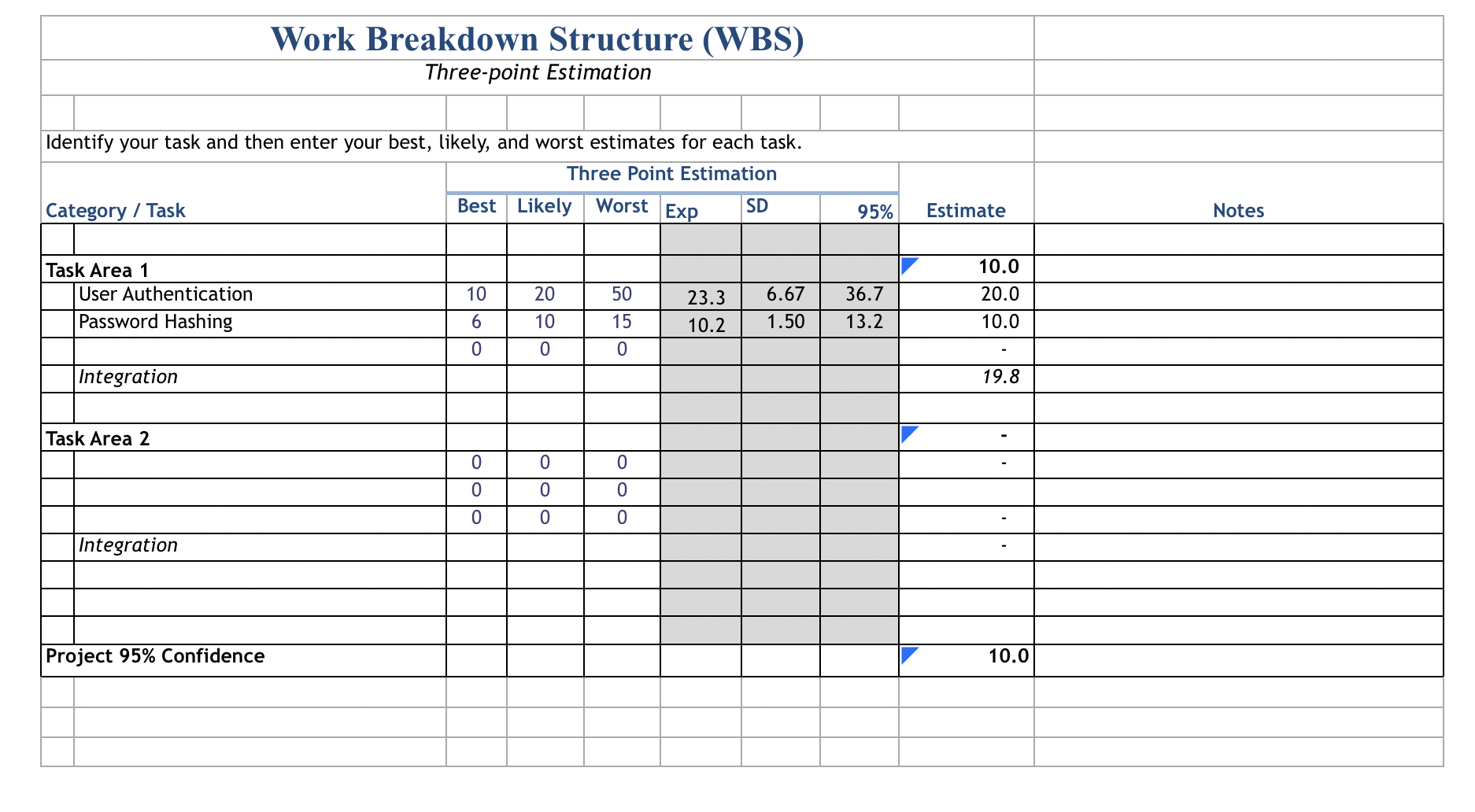
* Sprint planning following the scrum framework
* Ego-less workflow where task oversight is done by everyone
* Documentation Stored in Overview > Sub-App directories

Tools

* UML: [draw.io](http://draw.io)
* Database: Posegres
* Server: Heroku
* Programming Languages / Frameworks: Node.js, Jquery, HTML, CSS, PSQL, PLSQL, Vuejs or Angular? Undecided

# Risk Analysis

Initial risks are task specific with our initial goals of getting a login page and user authentication setup. The time estimates for this are in the table below. We assumed where this was something that we have not done before they are overestimated where we will likely run into hangups.



Type to enter a caption.

# Configuration Management

See the README.md in the Git repository.