

Smoke

Project Outline:

In order to complete the project, as a group we decided to prioritize a balance of client desires and functional requirements that will elevate Smoke as much as possible. To exemplify this, we ordered our user stories in a top-down approach with the more necessary items at the top of the list. These are tasks and ideas that we will put the most effort in completing. The items toward the end of the list aren't as necessary and if they are not completed within the allotted time, the leftover features could be added at a later time.

With our upcoming sprints, we will continue to add details to our UML class diagram for our Suite of Games. Furthermore, it is imperative that we decide on what frameworks we will be using and how we will go about our front end and back-end development. From there we will complete tasks and fulfill user stories as they are prioritized on our list. We will divide that work evenly amongst each other and work together to complete as many tasks as we can.

Sprints:

For each sprint, as a group we will hold a discussion at the beginning of the sprint to decide what work needs to get done for that specific sprint. Throughout each sprint we will continue to hold standups/meetings where we will update each other on the work that we have done, so that everyone is on the same page. At the end of each sprint, we will recap our work and begin our transition to the next sprint.

User Stories:

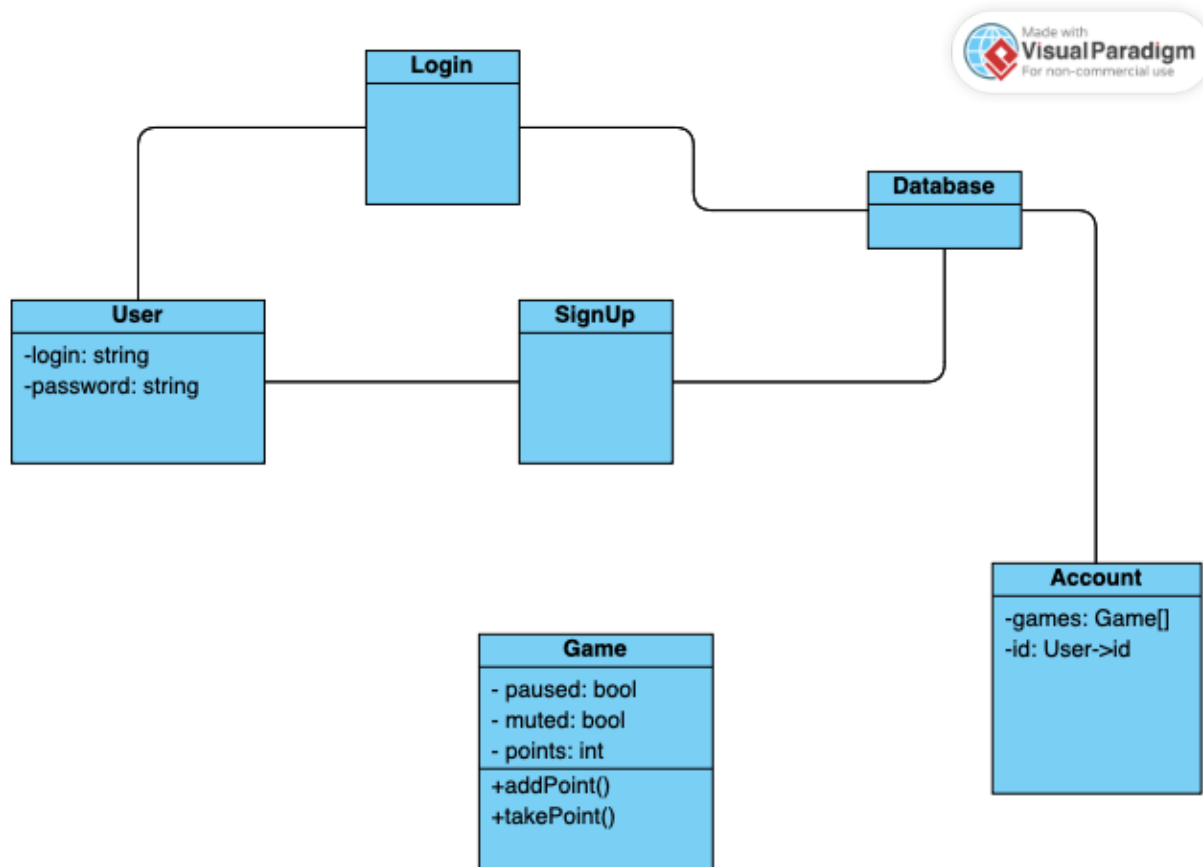
- As SOG I want a name better than “Muke-Us3”, so users will like the website better. (1)
- As a user, I want the SOG’s to use version control system to manage changes to the codebase and to facilitate collaboration among developers, so that it is easier to keep track of changes and make it easier to roll back changes. (1)
- As a SOG’s developer, I want the SOG’s to use standard interfaces and protocols, so that I can communicate between different components. (2)
- As a SOG developer, I need it to be created in a way that allows for new games and more users to be added later without slowing things down or causing problems with the system. (8)
- As a SOG’s developer, I need to make sure the html GUI code does not contain any business logic or game logic, so that the client can plan for the mobile market. (2)
- As a SOG’s developer, I need to make sure the html GUI code does not contain any business logic or game logic, so that the client can plan for the mobile market. (2)
- As a user, I want to communicate between different components, such as game engines, user interfaces, and databases, so that it is easier to swap out components or to integrate with third-party tools. (4)
- As a SOG’s developer, I want to create a concrete implementation that returns fake data, so that I can test certain aspects of the database. (4)
- As SOG, I need the games to be implemented as components so we can replace them seasonally. (8)
- As SOG I want to be able to play some of Steven Lamberts games on a website, so my users can access multiple games at one time. (2)
- As a SOG’s dev, I need to make sure all game assets are less than 100k in size, so that they load quickly over the internet. (1)
- As a user I would like to access multiple games at once, so I don’t get confused by multiple pages. (4)
- As SOG, I want it to be easy to select and navigate to games, so my users will stay engaged and explore the website. (4)
- As SOG, I need there to be minimal crashes, so I do not lose users. (8)
- As a user, I want error messages and screens to be funny, as well as telling me what went wrong, so I can laugh and feel better when something goes wrong with the software. (2)
- As a SOG developer, I need the user to log in so we can save their information to display later. (8)
- As an administrator, I want an admin login name of SoggieMuffins and password of 1234, so that they have an easy way to change the duration of ads. (1)
- As a customer I would like to be able to make my own username and password, otherwise I will forget my credentials. (4)
- As SOG, I need for past users to automatically be logged in, so they will want to use my site later. (4)
- As a user I want to be welcomed when logging in, so I know that got onto the site correctly. (2)
- As a client, I want all passwords to be salted and hashed (using SHA) with the result stored in the database, so that customers and admins can login securely. (1)
- As a SOG user I would like to be able to just sign in using my FB/Google account, because I always forget my credentials. (8)

- As a user I would like to see my stats and games on the same page, so I don't get confused with multiple pages. (2)
- As a user, I want my stats always displayed, so I know when I beat my high score. (2)
- As a user, I want the dashboard to expand the games when selected, so I don't have to switch pages. (8)
- As a user, I would like to be able to use just one hand to play, so I don't get bored with the games. (2)
- As a user, I want to be able to pause the games, so if I need to do something else, I don't lose. (2)
- As a user, I would like to be able to save my progress and come back later, so I can show my friends how far I got. (4)
- As a user, I want lots of different colors on the website, so I stay engaged with the website, and so it captures my attention. (2)
- As a SOGGIE rep, I want the users to have different "skins"/looks to choose from, so that the games can be especially appealing to women. (4)
- As a user, I want to have different "skins"/looks in the games, so that I can better express myself and feel a part of the characters. (4)
- As a user, I want medium-quality graphics and sound effects, so that I can be more engaged and immersed in the games. (1)
- As a user, I want the sounds to be functional as opposed to decorative, so that the sounds are more engaging and indicated events in the game. (1)
- As a user, I want the ability to mute sounds and have those sound settings saved from one session to another, so that I mute certain sounds and not have to reapply those settings in every session. (4)
- As a SOG developer I would like to have the user's information stored in a database, so we can keep track of their scores over time. (4)
- As a SOG's developer, I want to use the singleton pattern, so that I can keep the database code in one place. (8)
- As a SOG's developer, I want to use the bridge pattern to create a database manager, so that the database manager can communicate with a database implementation interface that has numerous concrete database implementations. (8)
- As a SOG dev, I need to create a bridge to a potential API with a stub that will deliver simple ads, so that the client can monetize their games. (8)
- As a client, I need the ability to display adds, so that I can make money. (8)
- As a younger player I need a tutorial, so when I don't know what to do, I can continue enjoying the website. (1)
- As a player I want to discover a hidden easter egg, so there is an element of surprise to the game. (1)

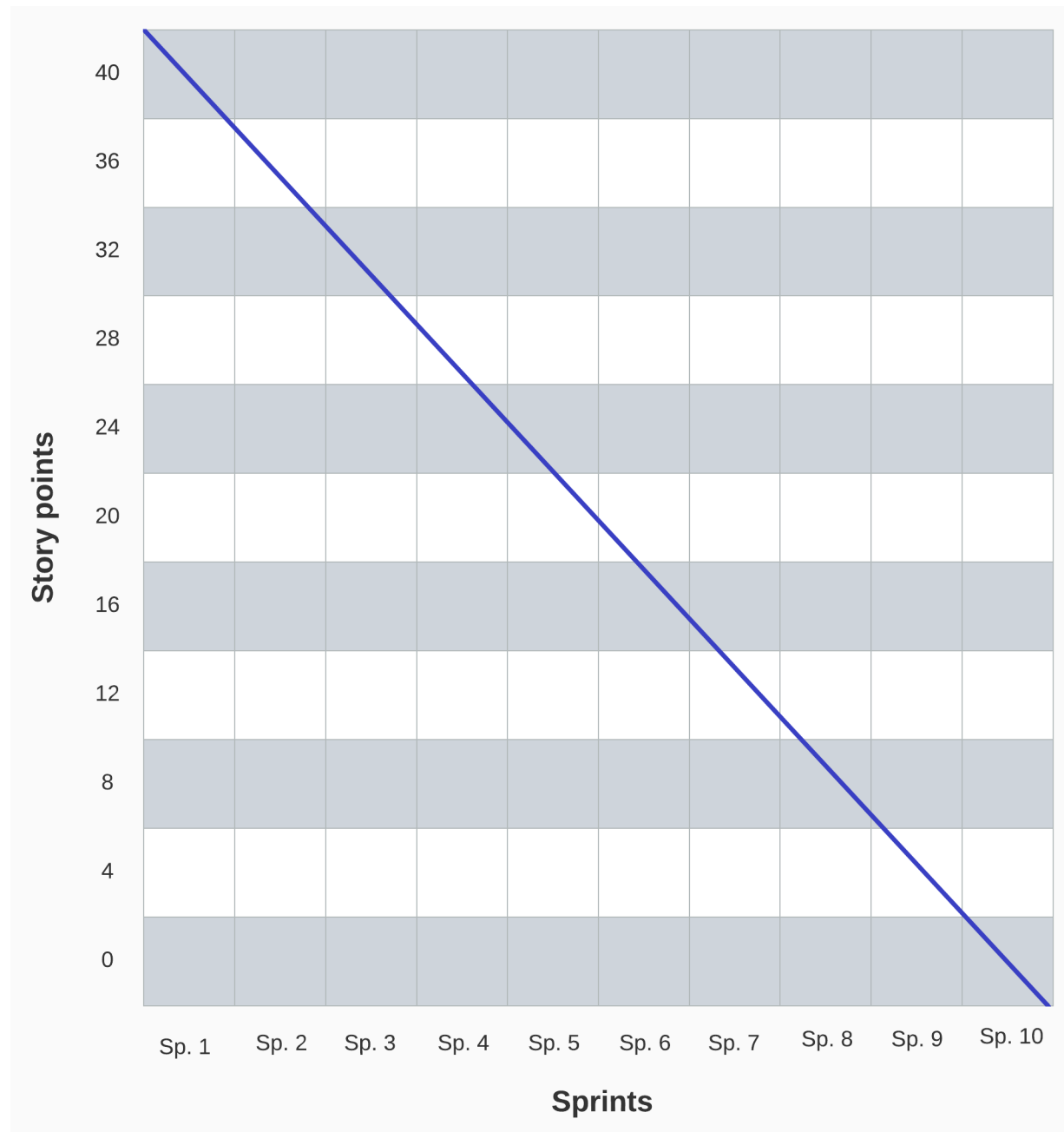
Task List:

- A new name that the users will like not "Muke-Us3"
- use version control system for coding
- use standard interfaces and protocols
- scalable coding in mind
- html code does not contain business or game logic (for mobile market) -coding with compatibility with different components and technologies in mind -make fake data to test the website
- make different games accessible at once
- make sure the game assets are less than 100k size
- minimal bugs and crashes
- add funny error messages
- user information to be saved on login
- admin login with "SoggieMuffins" username and 1234 password
- allow for users to make their own credentials to log in
- make it so users stay logged into the site
- add welcome landing page
- securely store passwords
- FB/Google log in options
- stats on game pages for users
- make sure stats are always displayed
- expandable dashboard
- games coded for just one hand gameplay
- add pause screen
- be able to save progress
- make website graphically appealing
- different skins/looks for the games
- medium quality graphics and sound effects for the games
- ability to mute sounds in settings and they save
- user data stored in database (high scores)
- use a singleton pattern
- use a bridge pattern to make a database manager
- create a bridge to a potential api to deliver ads
- add a tutorial for some users
- add an easter egg for the user to find

UML Diagram:



Burndown Chart:



<https://github.com/maxzinzi/FinalProjectCS325>

Joshua Marion, Maxwell Zinzilieta, Miles Wilson