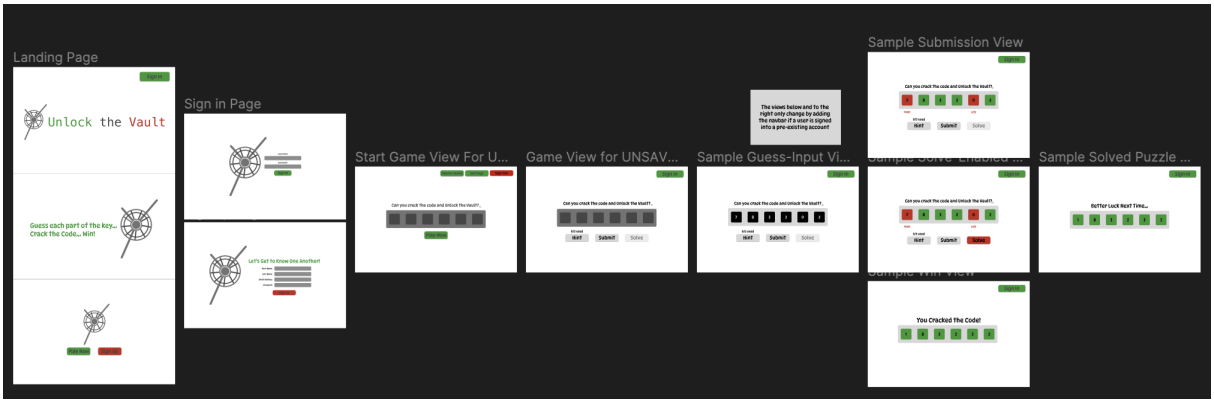
Product Design Specification: Unlock the Vault

Project Description

Product Description

- This will be a guess-related game where a unique 6 digit code will be generated every game for the user to guess.
    - Users get up to three hints, each of which will check the guesses against the correct code, and tell the user if the each digit in each spot is less or more than the correct digit for said spots, or correct.
    - After the user runs out of hints, they will be able to click the solve button to see the solution.
    - The submit button only indicates if an answer is correct or not for each spot.
- Wireframes:
    -



Methods

- **submitCheck()**: Will compare the digits in each spot against the correct digits in the unique code. The background color for each cell will turn to red or green depending on whether or not they got it wrong or right respectively.
- **hintCheck()**: Will compare the digits in each spot against the correct digits in the unique code. The background color for each cell will turn to red or green depending on whether or not they got it wrong or right respectively. Will also display whether the incorrect guesses are greater than or less than the correct ones for each spot.
- **activateSolve()**: Checks how many hints the user has asked for. If hints == 3, then the solve button will be activated.

Market Space

- This game is simple to learn, yet still challenging enough to be consistently engaging.
- Since there are no levels nor a consistent answer, the game avoids becoming redundant.
- Educational: While it may seem a generally simple guessing game of chance, younger kids can use this to reinforce basic concepts like greater/less than that they'd encounter in a math class.

Functional Specifications

Features

- **User Authentication**
    - Sign Up
    - Log In
- **Game**
    - Users can start a game with or without an account.
    - If the user has an account, data will be saved to it.
    - Each user gets up to 3 hints per round.
    - Users can request the solution after the 3 hints are exhausted.
- **User Dashboard**
    - Users can view statistics from past games including how many hints were used and whether or not they won.

Deployment

- I will deploy the Flask project with Heroku.
    - After creating a Heroku account and installing the Heroku CLI, I can create a file named Procfile in the project's root directory, then updating the requirements.txt file before committing the files to git.
    - Next, I'll use the CLI to create a Heroku application, then pushing it to a remote repo created with the Heroku "create" command.
    - Finally, the project will be built and deployed, resulting in a viewable app in the browser.

Upcoming Milestones

M1: 2/4 - 2/16: GitHub setup, Flask project blueprint, finalize website design
M2: 2/17 - 2/29: User authentication (account creation, login/out)

M3: 3/1 - 3/20: Game functionality
M4: 3/21 - 4/10: Work on/Refine UI
M5: 4/10 - 4/15: Dashboard → Ensure user can view data from previous games if they have an account
M6: 4/16 - Finals: Testing, project refinement