**Infinite Wordle**

**Fabi**

# PROJECT PROBLEM STATEMENT

Wordle is fun, but the official game limits you to one puzzle a day. My project is an Infinite Wordle app where users can play unlimited Wordle puzzles anytime. The problem I’m solving is making a fresh, endless Wordle experience with smooth gameplay and feedback on guesses.

# CONCISE DESIGN OVERVIEW

I’m using SwiftUI for the user interface because it works well on iOS and makes animations easy. The game logic will handle generating random words, checking user guesses, and updating the colors of letters based on accuracy. The main components are the game board view, keyboard input, guess validation, and a scoring system.

# SCOPE STATEMENT

* Unlimited Wordle puzzles with random words
* Color-coded feedback for guesses (correct spot, wrong spot, not in word)
* Keyboard for user input
* Tracking guesses per puzzle  
  **Does not include:**
* Multiplayer or sharing puzzles
* Daily challenge mode
* Advanced AI hints or word suggestions

# TIMELINE

Draw a timeline, including milestones to serve as the basis for a work breakdown structure (WBS) and appropriate Gantt or PERT charts. For this class, you can refer to the Work Breakdown Excel Sheet. This can be a table with anticipated tasks listed for each school day of the project. You will submit completed chart(or Record of Tasks) along with your final deliverables.  
  
SEE CHART

# TEST PLAN

# I’ll test the game mostly by playing it myself on simulator and device:

# Enter valid and invalid guesses (manual)

# Check if color feedback matches the guess results (manual)

# Restart puzzles and make sure new words load (manual)

# Try edge cases like guessing too many times or empty input (manual)

# No automated tests planned due to small scope, but I’ll log bugs and fix them as I find them.RISKS AND CONTINGENCY PLANS

* List known and potential risks by estimated probability, with mitigation plans.  
    
  **Word list issues** — keep backup word lists and test word loading thoroughly
* **UI glitches or crashes on keyboard input** — add input validation and test extensively
* **Running out of time to add restart feature** — deliver core gameplay first, add restart later if needed
* **SwiftUI limitations on animations or layout** — simplify UI or switch to UIKit components if needed