Wide Words Contents

Wide Words Contents

Description

Intended User

Features

User Interface Mocks
- Home Screen
- Flashcard Screen
- Quiz Screen
- Quiz Completion Screen
- Homescreen Widget
- Settings Page
- Key Considerations
  - How will your app handle data persistence?
  - Describe any corner cases in the UX.
  - Describe any libraries you'll be using and share your reasoning for including them.
  - Describe how you will implement Google Play Services.

Next Steps: Required Tasks
- Task 1: Project Setup
- Task 2: Implement UI for Each Activity and Fragment
- Task 3: Set Up Models and Content Provider
- Task 4: Populate Database with Data
- Task 5: Set Up Adapter and Cursor Loader in Quiz
- Task 6: Implement Chronometers
- Task 7: Tablet UI
- Task 8: Implement Google Play Services
- Task 9: Implement Sharing Intent
- Task 10: Set up Web Service
- Task 11: Data Download Intent Service
- Task 12: Homescreen Widget
- Task 13: Device Testing
- Task 14: Generate Signed APK

Appendix 1: Material Design Palette

Appendix 2: Possible Future Developments

GitHub Username: rustynailor

Wide Words

Description

The writer Will Self is renowned for deploying his vast, esoteric vocabulary throughout his novels and newspaper columns. A self described sesquipedalian (lover of obscure language), he argues persuasively that we could all benefit from being less risk-averse in our choice of words. This app is designed to help the user do just that - embrace a wider world of words.

Problem:
Will Self writes columns for many newspapers and magazines in the UK, which are peppered with obscure words. The user enjoys reading these articles, but finds having to constantly reach for a dictionary to look up words disrupts the flow of their reading. They want to learn and retain some of the words so they can read with more confidence and pleasure.

Proposed Solution
After analysing a range of Will Self columns, I have identified 50 uncommon words that frequently occur in his writing. This app will help the user learn those words with engaging presentation and a simple but compelling multiple choice game.

The app will have two modes - a learn mode, where the user is presented with the words as flashcards with a full definition, which they can review and learn at leisure.

Then there is a game mode, where the user has to select the correct definition of a word from four choices within ten seconds. These quizzes will go through batches of ten words at a time. Once a user has successfully selected the correct definition of a word three times (in different quiz sessions), they have mastered that word, and it will not be presented again in the quiz. The object of game mode is to have mastered all 50 words, at which point the user should be able to read a Will Self column with vastly increased ease and pleasure.

In learn mode, there are three choices of review - all words, new words or mastered words. By selecting new words, the user can review only words that they have not yet mastered.

There will also be an option in settings to reset the list of mastered words.

Data will be persisted using a SQLite database. Sharing intents will be used to enable users to proudly display their achievements on Social Media.

Defintions for words will be sourced from the open source WordNet project:
http://wordnet.princeton.edu/

Intended User

The intended user of the app is an English speaking reader who is keen to expand their vocabulary with some interesting new words, or reinforce their knowledge. It will be of particular interest to fans of Will Self, but also to general readers.

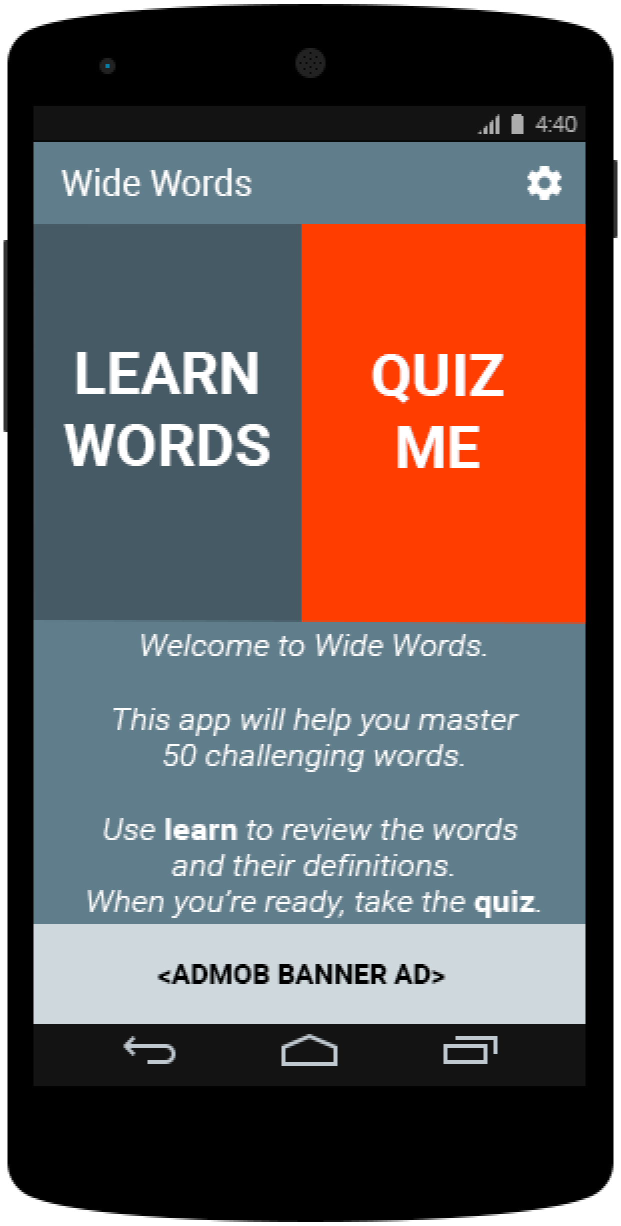
Features

The main features of the app are:

- Contains a SQLite dictionary of words with related definitions
- Presents these words to the user with the correct definition in flashcard format in learn mode
- Has a timed multiple choice quiz in Quiz mode
- Saves user progress on the quiz and modifies questions accordingly
- Displays ads on the home screen using Google Admob
- Has google analytics embedded in the app
- Allows the user to share their progress with a sharing intent
- The app supports Android 4.2 and higher
- A word of the day widget which shows a random word and definition
- An option to download the latest word list from a web service in the settings, via an Intent Service

User Interface Mocks

Home Screen



Welcome to Wide Words.

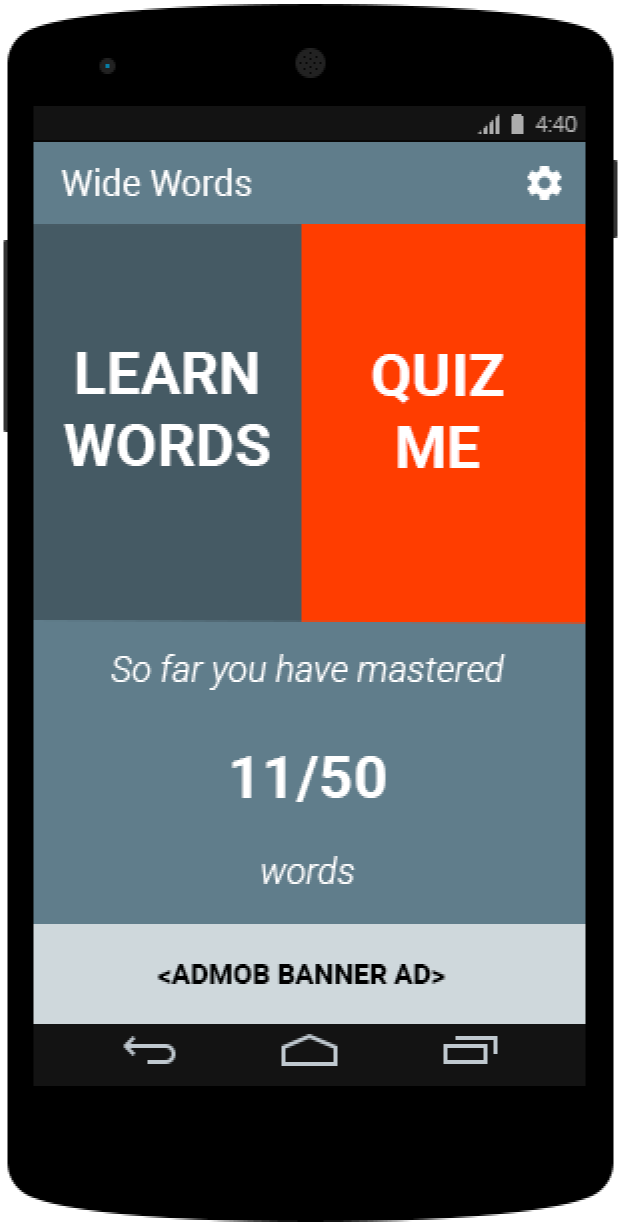
This app will help you master 50 challenging words.

Use ***learn*** *to review the words and their definitions.*
When you're ready, take the ***quiz****.*

This is the home screen of the app.
There is a toolbar at the top, with a settings menu icon option.
Below that, there are two options - start learning, or start quiz.
Underneath that there is some introductory text to welcome the user to the app and provide some instructions.

Finally, there is a Google Admob banner ad.

Once the user has successfully answered a question in the quiz, the introductory text is replaced by a message showing their progress:

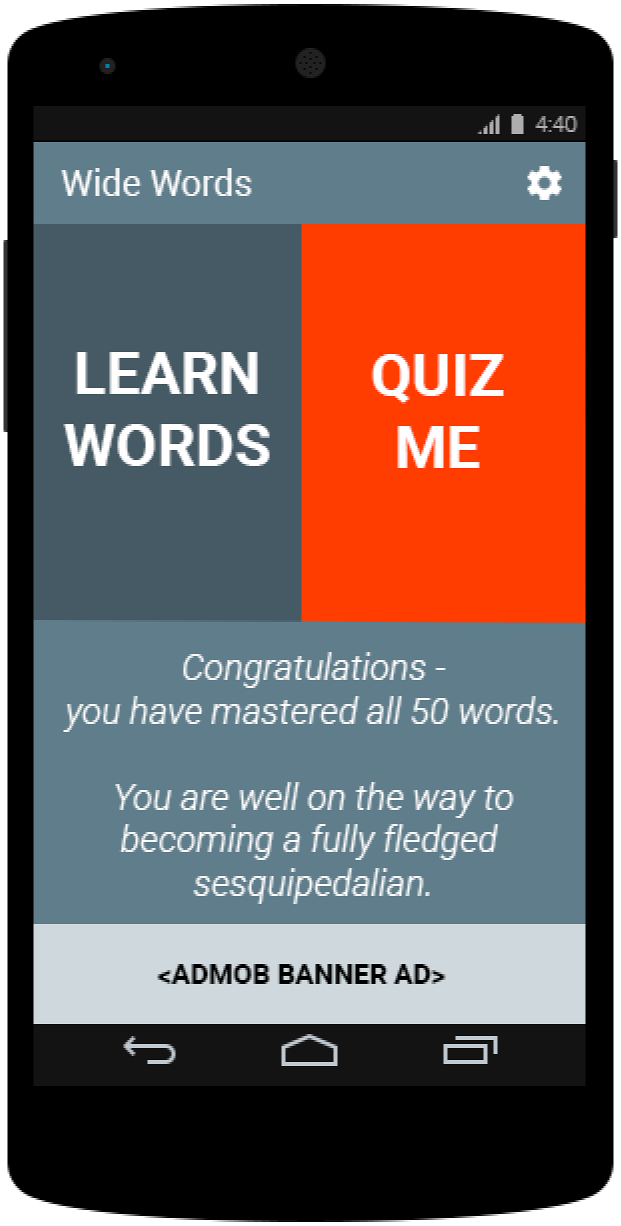


So far you have mastered

11/50

words

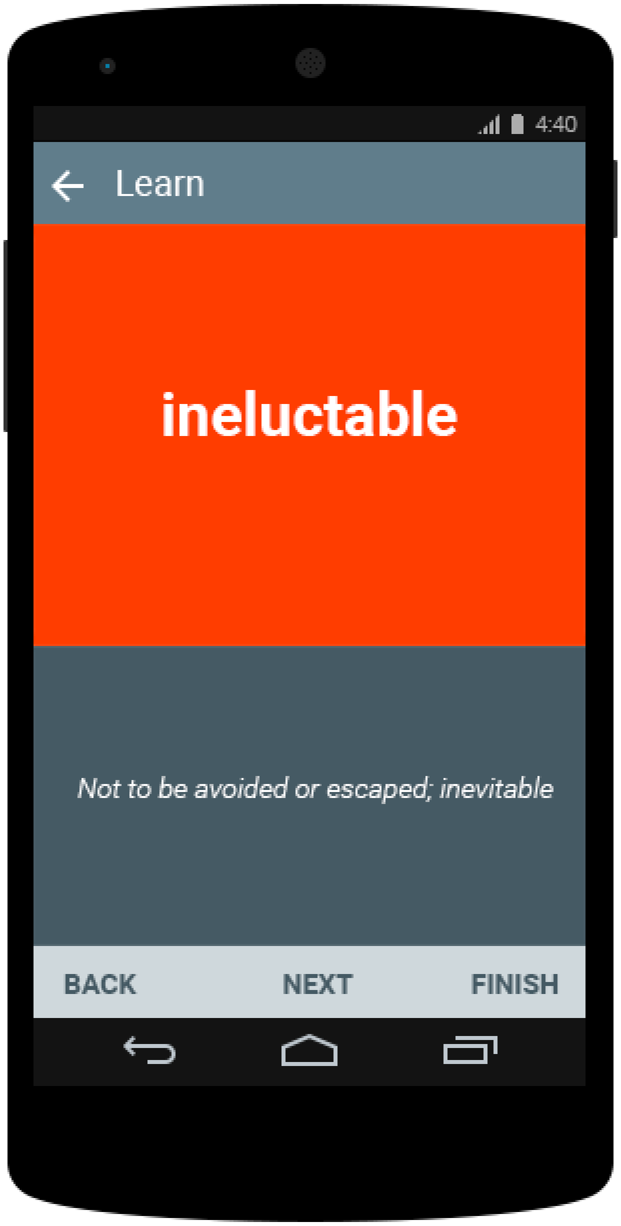
When the user has mastered all words, this is replaced with a congratulations message:



Congratulations - you have mastered all 50 words.

You are well on the way to becoming a fully fledged sesquipedalian.

Flashcard Screen



ineluctable

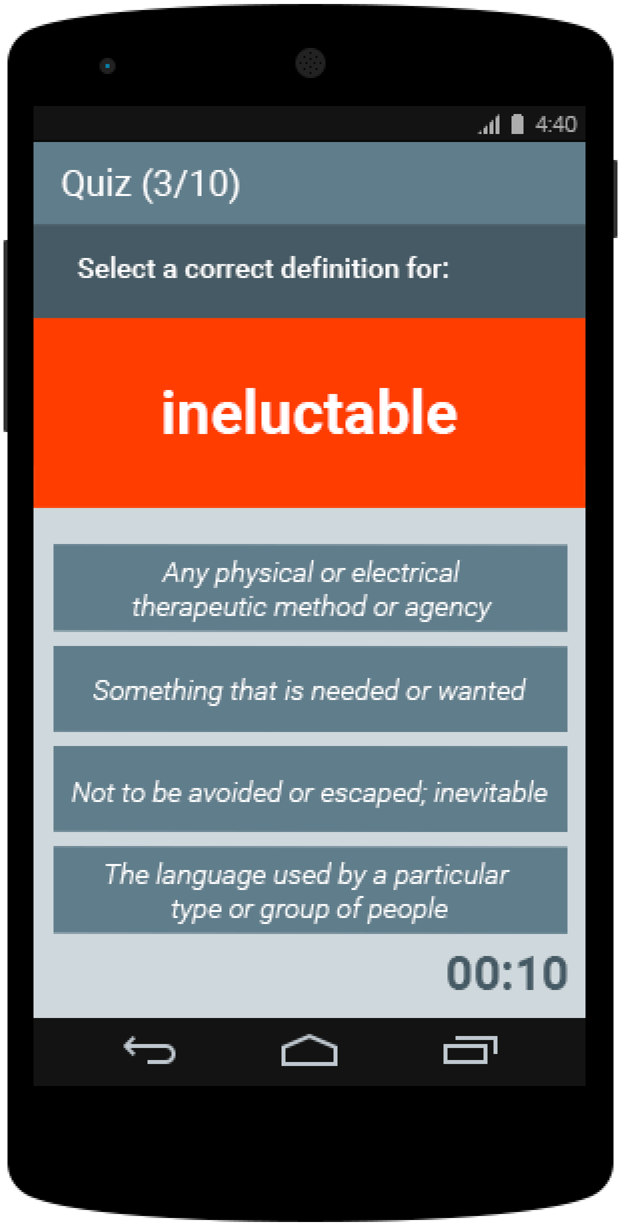
Not to be avoided or escaped; inevitable

In learn mode, the screen displays the word at the top, then a definition below.
The design is intentionally simple to focus the user's attention.

Back and next buttons will load the next word (if it is the first word, back is not shown. If you are at the last word in the set, next is not shown).

Finish exits the activity, as does the back arrow in the toolbar.

Quiz Screen



Quiz (3/10)

Select a correct definition for:

ineluctable

Any physical or electrical therapeutic method or agency

Something that is needed or wanted

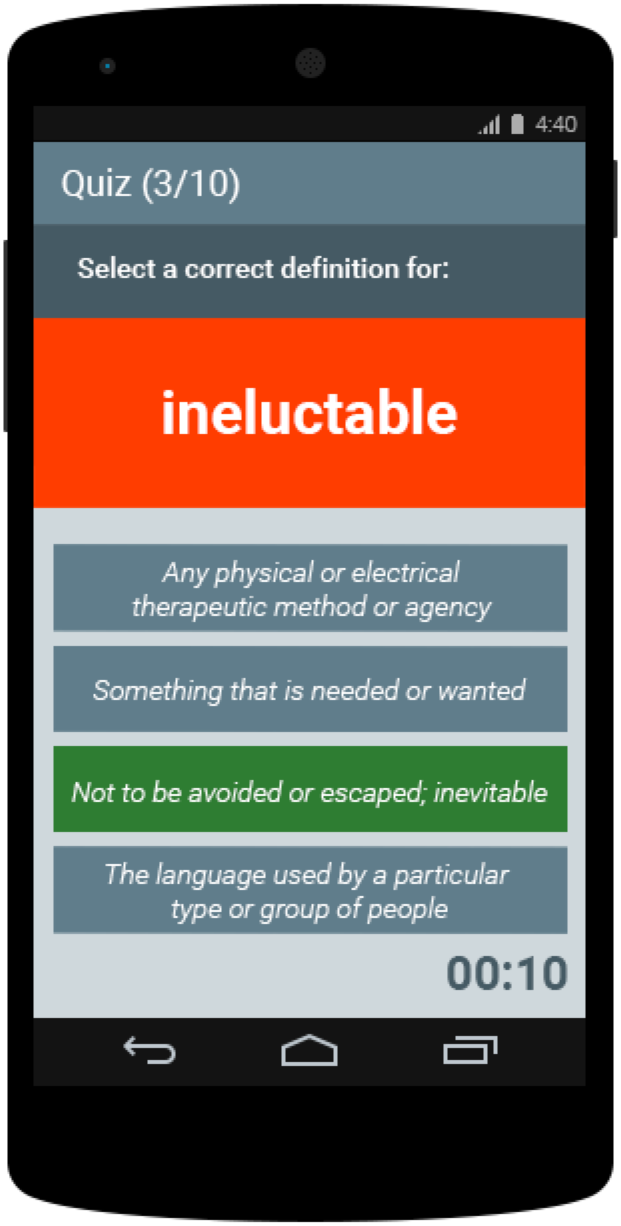
Not to be avoided or escaped; inevitable

The language used by a particular type or group of people

00:10

The quiz screen shows a choice of four definitions - the user must select the correct definition before the timer in the bottom right runs out.

On pressing the correct answer, the answer will briefly be highlighted in green before advancing to the next question:



Quiz (3/10)

Select a correct definition for:

ineluctable

Any physical or electrical therapeutic method or agency

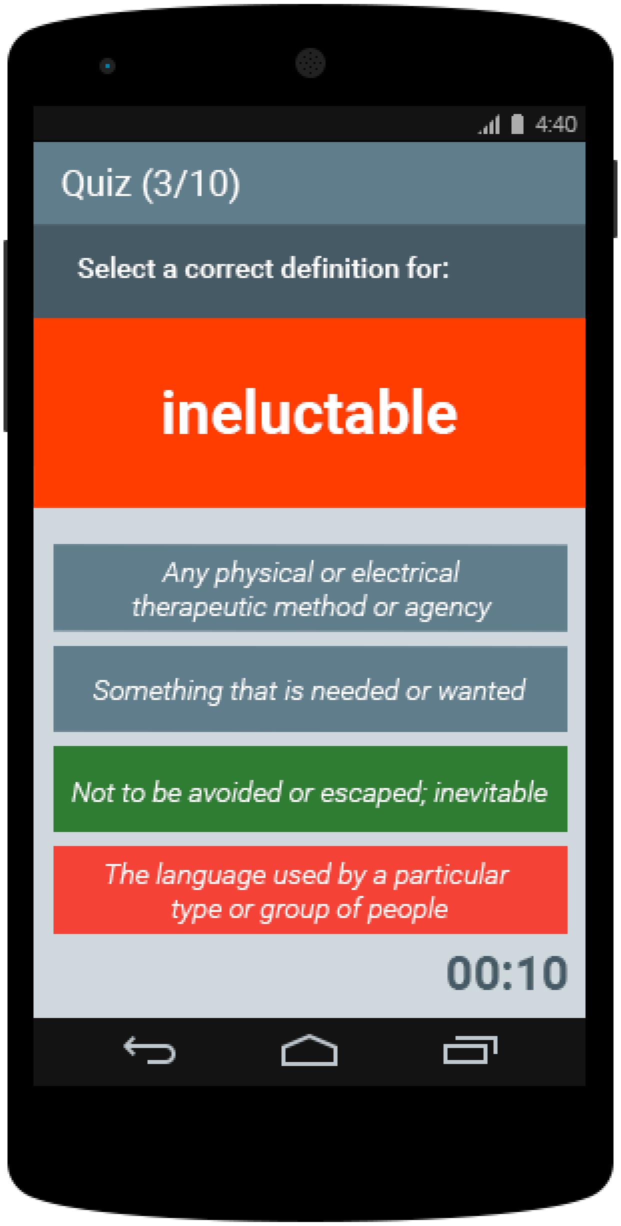
Something that is needed or wanted

Not to be avoided or escaped; inevitable

The language used by a particular type or group of people

00:10

If an incorrect answer is given, the user’s selection will be highlighted in red, and the correct answer shown in green for 2 seconds before advancing to the next question:



Quiz (3/10)

Select a correct definition for:

ineluctable

Any physical or electrical therapeutic method or agency

Something that is needed or wanted

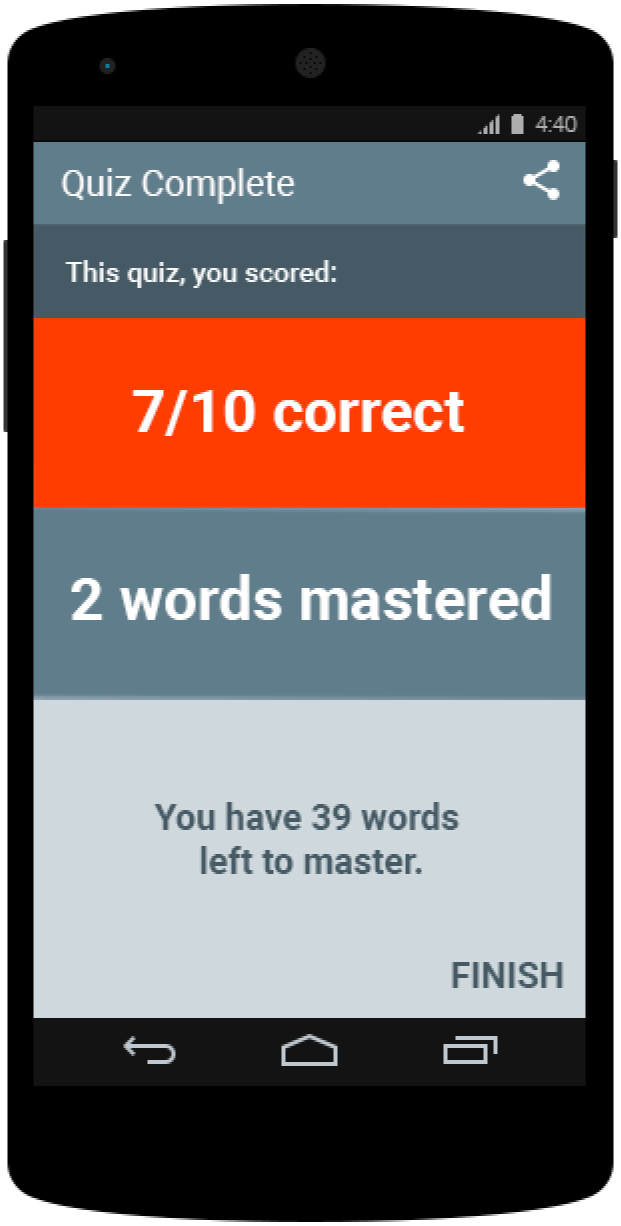
Not to be avoided or escaped; inevitable

The language used by a particular type or group of people

00:10

If the time runs out before the user answers, the correct answer will be highlighted and displayed for 2 seconds.

Quiz Completion Screen



4:40

Quiz Complete

This quiz, you scored:

7/10 correct

2 words mastered

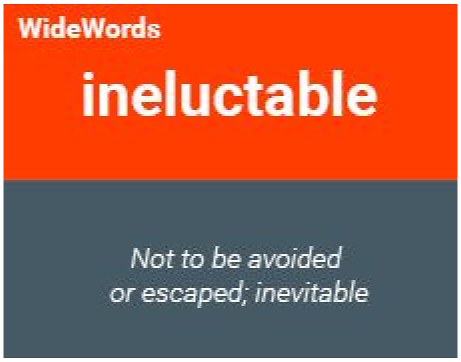
You have 39 words left to master.

The quiz completion screen shows the number of words answered correctly, the number of words mastered (answered correctly 3 times) this quiz, and the total number of words mastered so far.

A social sharing icon in the toolbar allows users to share their achievement (as text).

If the user has mastered all 50 words, a special congratulations message will be shown.

Homescreen Widget



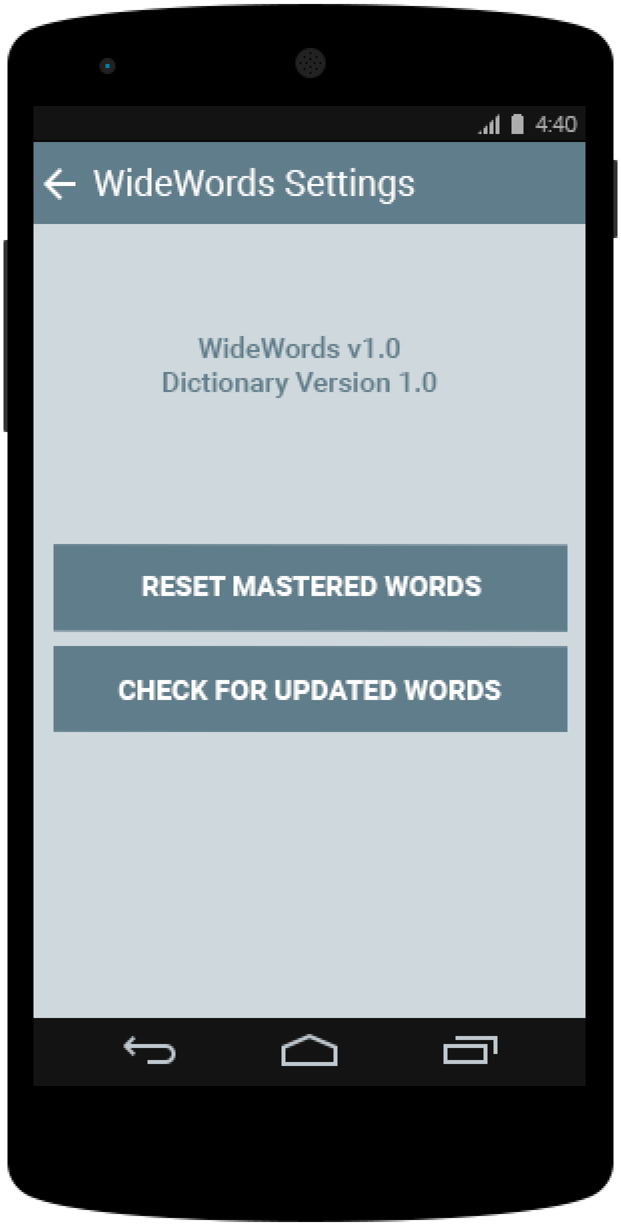
WideWords

ineluctable

Not to be avoided
or escaped; inevitable

This is a widget design to show a word and definition on the user’s home screen.

Settings Page



WideWords v1.0
Dictionary Version 1.0

The settings page shows the version number of the app and dictionary and presents two options.

Reset mastered words clears the database of mastered words, resetting the app to first installed state.

Check for updated words initiates an API request (using an Intent Service) to a custom wordlist endpoint. If the endpoint has a newer list of words, those are imported into the app.

No words will be removed from the app, only added - so user progress will be retained even as new word lists are made available.

For testing purposes, an endpoint will be created that adds ten new words to the app (taking the word total to 60) if this option is selected.

Key Considerations

How will your app handle data persistence?

Data will be persisted in a local SQLite database, with a Content Provider used to access and update the data (which will contain a list of words and track the user's progress).

Describe any corner cases in the UX.

If a user exits a quiz without completing it, or is interrupted during a quiz, the quiz state will be persisted and they will have the option to return to it or start a new quiz when they click start quiz .

If the user has successfully mastered all the words, and they click quiz, a dialog will ask them if they would like to reset the mastered words (the same option as is in the settings library).

If there are fewer than ten words left to master, the quiz will consist of as many words as there are left.

In learn mode, once the user has mastered some words, they will be presented with a choice before they start looking at flashcards of either:

- All words
- Only words not yet mastered
- Only mastered words

If the user has mastered none or all of the words, they will go straight to the flashcards.

Describe any libraries you'll be using and share your reasoning for including them.

Schematic will be used to assist in the generation of a content provider, reducing the amount of boilerplate that is required (https://github.com/SimonVT/schematic).

Describe how you will implement Google Play Services.

Google Admob will be used to display ads on the homescreen of the app.

Google Analytics will be included so I can gather valuable data on app usage, which will be used to optimise the app in future versions.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Set up Github repo and add blank Android studio project to it
- Add Google play services for admob and analytics to the app build.gradle
- Add Schematic to build.gradle

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Home Activity
- Build UI for Flashcard Activity
- Build UI for Quiz Activity
- Build UI for Quiz Completion Activity
- Build UI for Settings Activity

Task 3: Set Up Models and Content Provider

Build Content Provider and Related Models using Schematic Library

- Create a Models package and add POJO classes for Users, Words, Definitions, Quizzes and Quiz Results
- Create a Data package and create database and provider classes following the conventions of schematic
- Define end points (content URIs) in the provider class for Users, Words, Definitions, Quizzes and Quiz Results

Task 4: Populate Database with Data

- Add a seed method that will be called on database creation (that is, app installation)
- Populate the method with insert statements that will provide our initial dictionary of words

Task 5: Set Up Adapter and Cursor Loader in Quiz

- Create an adapter for the quiz activity to handle the loading of answers and definitions
- Add a Cursor Loader to the Quiz Activity to acquire this data from the database then pass the result cursor to an instance of the adapter

Task 6: Implement Chronometers

- Add logic to the quiz task so the countdown timer for a question is linked to an implementation of Android chronometer.
- The time taken to complete a question should then be stored in the database with the answer given

Task 7: Tablet UI

- Add an additional quiz layout file to support the tablet specific design of the quiz activity

Task 8: Implement Google Play Services

- Add Google analytics to the app
- Add an advertising banner to the home screen using Google admob

Task 9: Implement Sharing Intent

- On the completed quiz activity, add a sharing intent so the user can share their progress (in the form of a string which gives their progress in the app).

Task 10: Set up Web Service

- Set up simple web service to supply wordlist and definitions in JSON format
- Make available at test URL

Task 11: Data Download Intent Service

- Add an Intent Service to handle API based update of the word list
- Build appropriate POJO classes to model the JSON payload from our webservice and update the data model appropriately

Task 12: Homescreen Widget

- Build UI for word of the day widget
- Implement word loading from content provider

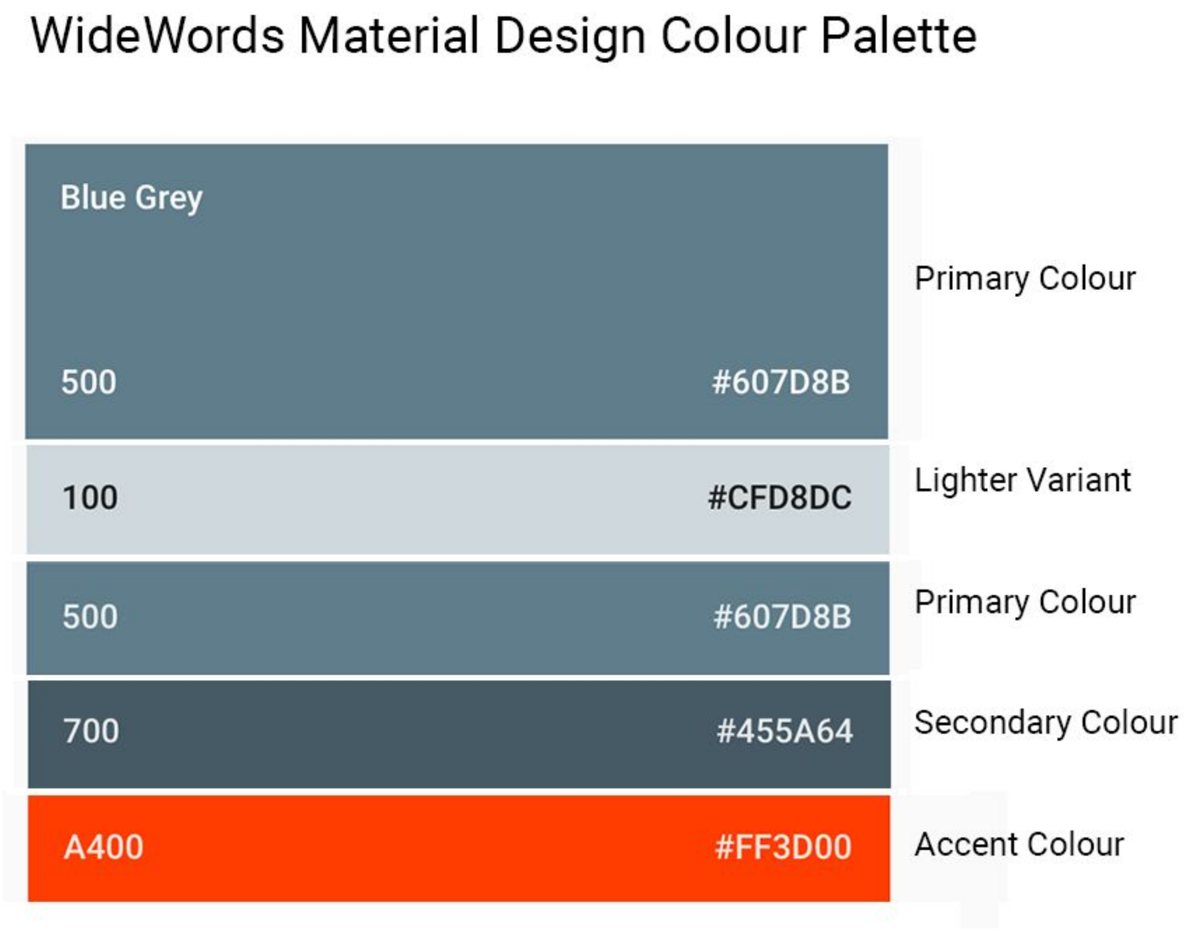
Task 13: Device Testing

- Install the app on a range of test devices, and check performance and UI consistency
- Remedy any issues found

Task 14: Generate Signed APK

- Generate signed APK and submit.

Appendix 1: Material Design Palette



WideWords Material Design Colour Palette

Blue Grey

500	#607D8B	Primary Colour
100	#CFD8DC	Lighter Variant
500	#607D8B	Primary Colour
700	#455A64	Secondary Colour
A400	#FF3D00	Accent Colour

Appendix 2: Possible Future Developments

If the app proves successful, there is plenty of scope to enhance it. Beyond the initial set of words that come supplied, future releases could have the ability to download bundles of words (themed around writers) so that there are new challenges once they have mastered the Will Self set.

There could also be further integration with Google Play services to provide a competitive online leaderboard, and additional sharing functionality related to that.

Additional types of games could also be added, to broaden the challenge - for example, selecting the correct word to use in context.