

How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
 2. Name your document file: “**Capstone_Stage1**”
 3. Replace the text in green
-

Description

Intended User

Features

User Interface Mocks.

Screen 1

Screen 2

Screen 3

Screen 4

Screen 5

Screen 6

Screen 7

Screen 8

Screen 9

Key Considerations

How will your app handle data persistence?

Describe any edge or 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 or other external services.

Next Steps: Required Tasks

Task 1: 项目配置和创建 Activities

Task 2: 为每个 Activity 和 Fragment 搭建 UI

Task 3: 实现主页面功能

Task 4: 实现单词清单和详情功能

Task 5: 实现设置功能

Task 6: 实现单词挑战功能

Task 7: 实现我的成绩功能

GitHub Username: FrankJenq

每天记单词

Description

这款应用可以帮助用户学习新单词，复习旧单词，还能通过单词测试挑战自我，轻松提升和巩固英文词汇量。

Intended User

想利用碎片化时间提升和巩固英文词汇量的英语学习者

Features

List the main features of your app. For example:

- Saves information
- Takes pictures
- Other features

在线加载和显示单词信息；

「记忆清单」将保存用户收藏的单词；

利用「记忆清单」保存的单词随机生成单词测试题目；

实时更新和保存用户的学习成就信息；

User Interface Mocks.

Screen 1



这是用户进入应用之后的主页面，会在线随机加载一个单词。单词下方显示了用户完成今日学习目标的进度。用户可以选择换一个单词，将单词加入记忆清单，或者直接点击「记住了」按钮，还可以进入单词挑战参加测试。如果出现网络问题，这里会显示提示信息。

Screen 2



用户可以点击主界面右上方的菜单，来进入记忆清单，查看成绩和修改设置。

Screen 3



用户在主页菜单栏点击「记忆清单」选项，会进入此页面，这里会以列表形式展现用户收藏的单词。用户可以通过按钮进行翻页，也可以点击列表中的单词进入单词详情页面。

Screen 4



在「记忆清单」中点击某个单词，会进入单词详情页面。这里可以点击「记住了」，这会将单词从记忆清单中清除，并计入用户记忆的单词总数。也可以点击「上一个」或者「下一个」按钮来切换单词。

Screen 5



点击菜单栏中的「设置」选项，会进入设置页面。这里可以输入每天计划记忆的单词数量，也可以在下拉菜单修改一次单词测试的题目数量，还可以选择是否开启学习提醒功能，如果选择开启，那么点击时间视图可以设置提醒时间。

Screen 6



点击主页面的「单词挑战！」按钮，会进入单词测试页面。这里可以选择测试包含的题目类型：根据含义选择对应单词，或者根据单词选择对应含义，或者进行单词与含义是否匹配的判断题。用户可以选择一到三种题目类型，测试题目会从「记忆清单」中随机选择单词来生成。

Screen 7



单词测试题目页面，这里显示的是测试题内容，根据题目类型的不同会有变化。

Screen 8



测试题结束后显示的页面，这里会显示用户本次测试的成绩，以及所有测试的平均成绩。用户也可以分享结果到社交媒体上。

Screen 9



点击菜单栏的「我的成果」按钮进入的页面。这里可以看到用户总共记忆了多少个单词，以及参加单词测试的平均得分等信息。用户可以把结果分享到社交

Key Considerations

How will your app handle data persistence?

使用 **Oxford Dictionary API** 在线获取单词数据；
 收藏的单词将使用 **SQLite** 数据库存储在本地；
 使用 **Content Provider** 来对数据库进行操作；
 用户的设置偏好和统计数据将通过 **SharedPreferences** 保存；

Describe any edge or corner cases in the UX.

假如记忆清单中没有单词，将在用户进入清单页面后显示提示信息；
 没有网络连接时会正确处理用户切换单词的动作，并显示提示信息；

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

使用 **Okhttp3** 来进行网络通信，这可以增加数据连接的稳定性；
 使用 **Retrofit2** 来在线获取 **JSON** 数据，从而提高数据获取的效率；
 使用 **GSON** 来生成 **JSON** 数据对应的 **POJO** 文件，这能够简化对 **JSON** 数据的处理；

Describe how you will implement Google Play Services or other external services.

集成定位和广告两项 **Google Play Service**，这样能够根据地理位置来展示广告。

Next Steps: Required Tasks

Task 1: 项目配置和创建 Activities

- 1.在 build.gradle 中配置 libraries
- 2.创建 Activities，目前预计需要创建的 Activities：
主页面-MainActivity
设置页面-SettingsActivity
记忆清单主页-ScheduleActivity
单词详情页面-WordDetailActivity
单词挑战主页-WordsChallengeActivity
单词挑战题目页-QuizActivity
我的成绩页面-MyGradesActivity
- 3.在 AndroidManifest.xml 中，设置需要的 permissions，注册 Activities

Task 2: 为每个 Activity 和 Fragment 搭建 UI

创建 UI 框架，为每一个 Activity 建立对应的 xml 文件，这一步完成之后，可以启动应用查看初始状态。

Task 3: 实现主页面功能

实现主页面的以下功能：

- 1.主页面在线获取单词；
- 2.添加单词到记忆清单，用 SQLite 存储数据；
- 3.通过「换一个」或者「记住了」按钮，切换主页面显示的单词

Task 4: 实现单词清单和详情功能

实现单词清单的以下功能：

- 1.从数据库获取记忆清单的内容，并分页面展示；
- 2.通过按钮切换上一页和下一页；
- 3.点选单词进入单词详情页面
- 4.在详情页面可以通过「记住了」按钮，从单词清单数据库删除这个单词。
- 5.记忆的单词总数会保存在一个 SharedPreferences 中

Task 5: 实现设置功能

实现设置页面的如下功能：

- 1.输入每天的学习目标，即计划记忆多少个单词

- 2.选择每次单词挑战生成的题目数量
- 3.修改后的设置会保存在 **SharedPreferences** 中

Task 6: 实现单词挑战功能

实现单词挑战页面的如下功能：

- 1.选择要包含的题目类型
- 2.根据 **Preference** 和题目类型，从记忆清单数据库选取数据，生成对应的题目
- 3.记录用户每道题的答题结果，最后统计出测试得分
- 4.在挑战结束的时候，显示本次得分和平均得分
- 5.实现分享功能
- 6.最终得分会保存在一个 **SharedPreferences** 中

Task 7: 实现我的成绩功能

实现我的成绩页面的如下功能：

- 1.从 **SharedPreferences** 中获取记忆的单词总数和测试平均分，并显示在页面上
- 2.实现分享功能

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"