

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](#)

[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: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: `varunam`

Kannada Riddles

Description

We are reaching those days where internet is becoming everything and people no more know about riddles/proverbs etc. This app will play role in taking riddles (as of now) and proverbs (may be in future) to the next generation through an app.

Intended User

Our intent users can be anybody who would like to spend time answering riddles. This might even be helpful for parents to make their kids aware of riddles and proverbs. This would help people kill their time in a meaningful and fun way.

Features

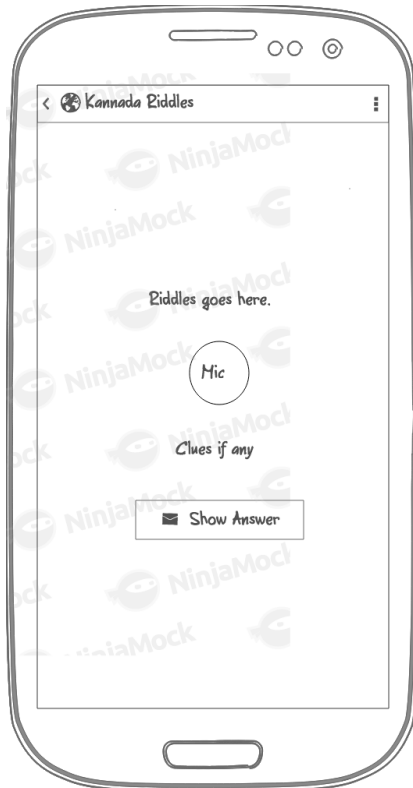
List the main features of your app. For example:

- Shows riddles with answers. Design would be similar to the news app Inshorts
- Every 10 swipes would give them score on how much they answered correct
- Each page will have a riddle, an option to speak out answer or type, clue (if any) and correct answers
- Score card will have an option to share it
- Users can also continue swiping to answer more riddles.

User Interface Mocks

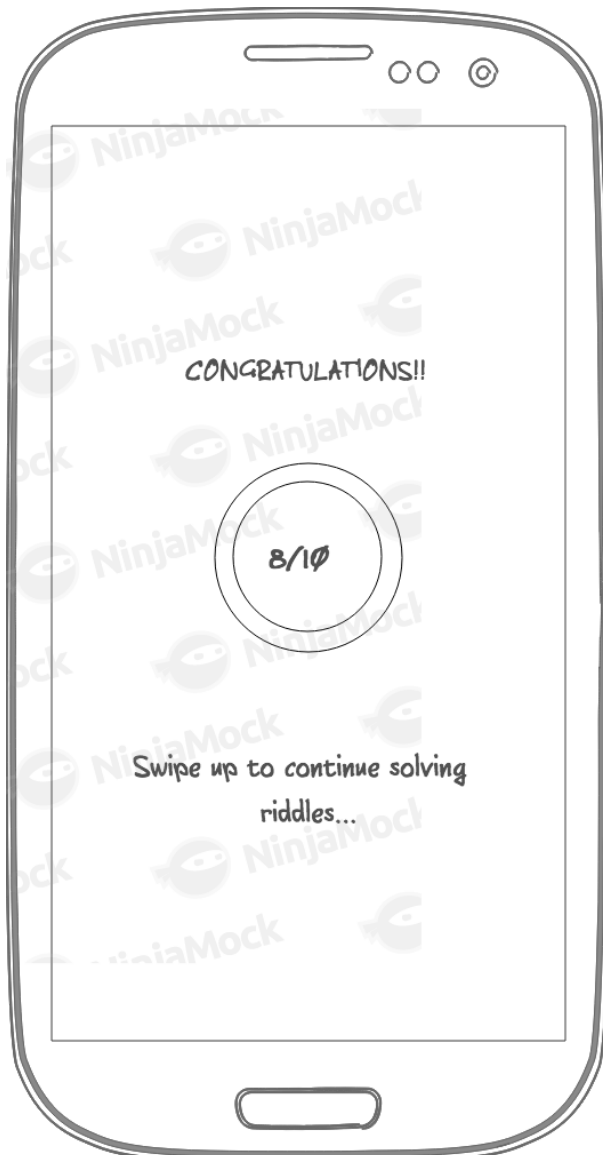
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



The above page shows mock of almost all the pages in the app. User will solve a riddle and swipe up to go to the next riddle. App will keep track of his progress

Screen 2



Once user solves a set of questions (may be 10) and swipe up the usual way, he will see his score of the session! The mock shows the same.

Key Considerations

How will your app handle data persistence?

The app will use Firebase Real time database to store riddles. They will be loaded run time and shown to the user

Describe any edge or corner cases in the UX.

If a user is solving a puzzle and presses back, an alert will be shown to him asking if he really want to quit solving puzzle and exit the app.

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

I might probably use libraries like Glide/Picasso for image and gif loading.

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

I will use Firebase Real Time Database for data persistence.

Database will have the list of riddles. App will load them during run time and show to the user. Currently, I am not planning to provide auth. All the user's progress will be stored locally using sharedpref.

Next Steps: Required Tasks

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

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks.

Connect project to Firebase Real time database

- Design Vertical Viewpager to provide "Swipe up to next Riddle" design.
- Implement Google Speech to Text api to provide speech input for riddle answers.
- Load data from Firebase database during runtime and show to the user.

Task 2: Implement UI for Each Activity and Fragment

- Build ViewPager for swipe up to next riddle actions.
- Build UI for results screens
- Build UI for special cards if any.

Task 3: Your Next Task

- Test UI and functionalities.

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**"