

## 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:** tryn2hard

# Pocket Tally

## Description

Pocket Tally is an easy intuitive way to keep track of your cricket game score. Best part is it's free!

## Intended User

Avid darts players, bar-goers, and everyday people who want to have a good time playing darts.

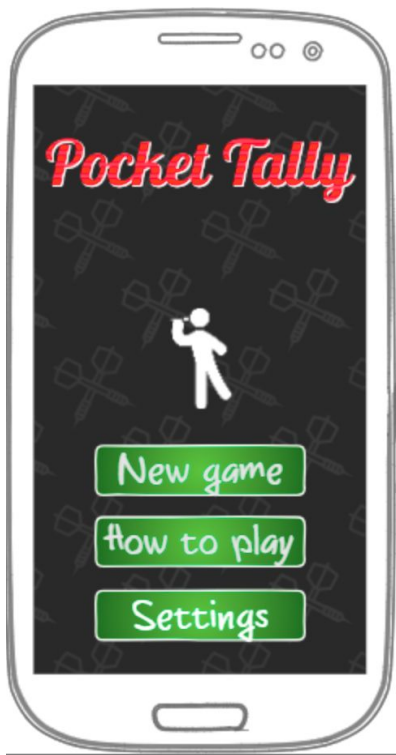
## Features

- 2 to 4 players in a game
- Standard or Cut-Throat game type
- Control the number of rounds in a game

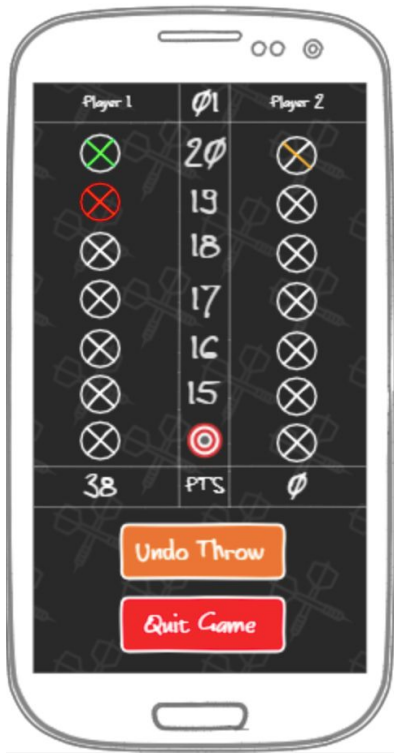
## User Interface Mocks

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](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

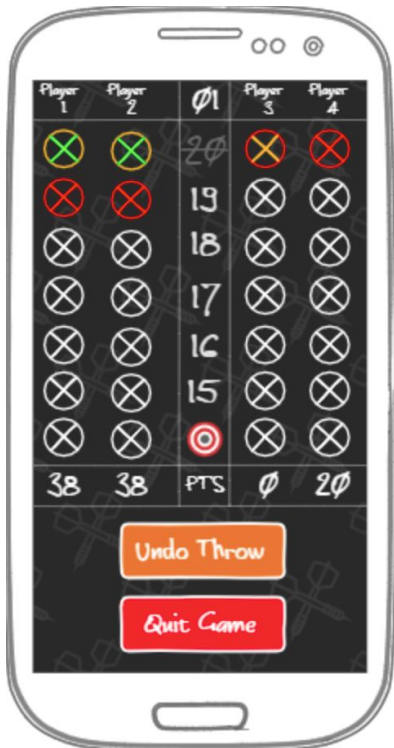
### Screen 1



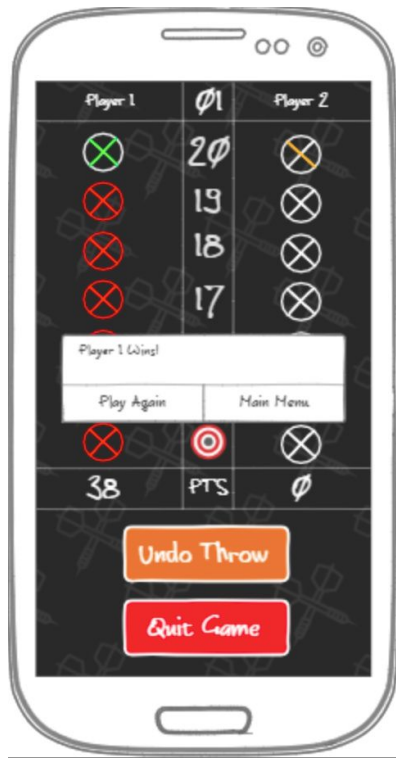
## Screen 2



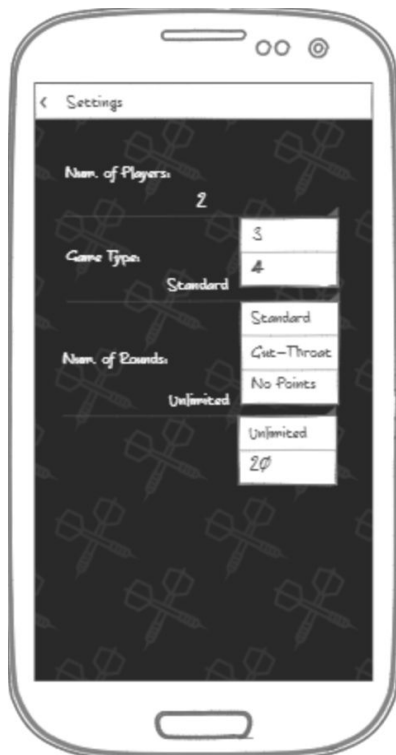
## Screen 3



## Screen 4



## Screen 5



## Screen 6



## Key Considerations

### How will your app handle data persistence?

The app will use preferences to store the user's settings. For each game of darts, the app will use Room to store the game data until the game is finished or ended.

### Describe any edge or corner cases in the UX.

In the case of the user leaving a current game, View Model will be used to make sure that the user can return to the game without the loss of any progress.

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

Room -- to store the relevant game data  
 Support Design -- to help create the necessary UI  
 ButterKnife -- to bindviews

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

Google Mobile Ads -- to have ads  
Google Analytics -- to have reports

## 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

- Create a github repo for the project and perform an init commit from AS
- Add all the necessary dependencies that will be used
- Configure the database

### Task 2: Implement UI for Each Activity and Fragment

- Build UI for the Starting Activity
  - Starting Activity will host three different fragments(Start Screen, Settings Screen, and How-to-Play)
- Build UI for the Dart Game Activity
  - Use fragments to change the number of players based on the user's settings

### Task 3: Connect the database to the Activity

- Use ViewModel and loaders to read/write to the database

### Task 4: Implement Google Mobile Ads

- Add an ad display to the Dart Game

### Task 5: Create a build variant for free and paid

- Create layout to remove the adds from the Dart Game

## Task 5: Write tests for parts of the app

- Create tests for the various activities

---

### 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**"