

## How to Use this Template

1. Make a copy [ File → Make a copy... ]
2. Rename this file: **“Capstone\_Stage1”**
3. Replace the text in green

## Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it **“Capstone Project”**
3. Add this document to your repo. Make sure it’s named **“Capstone\_Stage1.pdf”**

---

[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.](#)

[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:** akodiakson

# Pitch Counter

## Description

This app keeps track of the number of pitches a player makes during a baseball game. Pitchers are often at risk of arm injuries due to throwing too many pitches. This app will help you stay aware of pitch counts, as well as provide additional useful information like strikeouts, walks and hits, and how these numbers progress throughout a season.

## Intended User

The intended user is a spectator of a baseball game, often a baseball coach or a member of the pitcher's family.

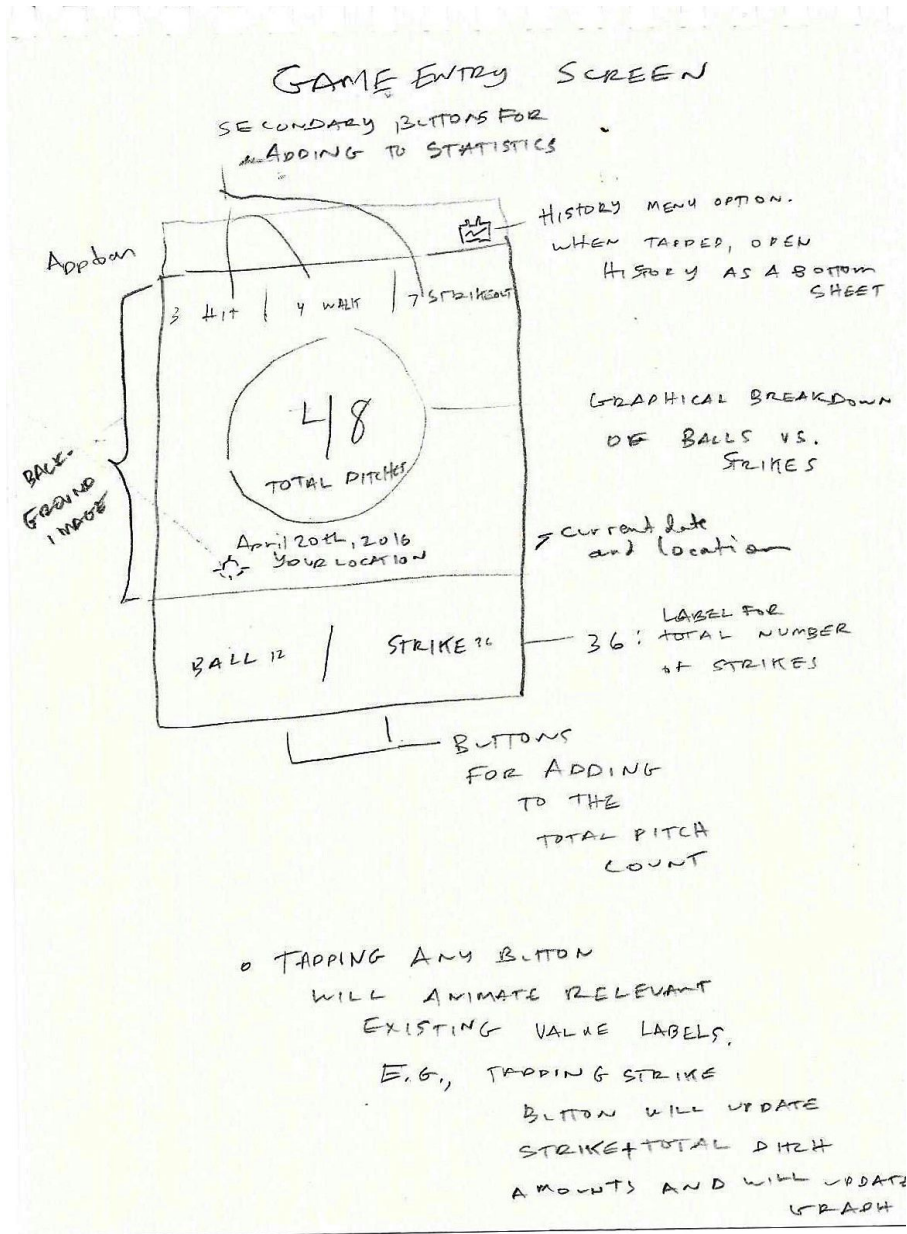
## Features

- Keep track of the number of balls and strikes thrown in a game.
- View history of previous games to see trends over a season.
- Track games through a widget.
- Add a current location to each game.

## User Interface Mocks

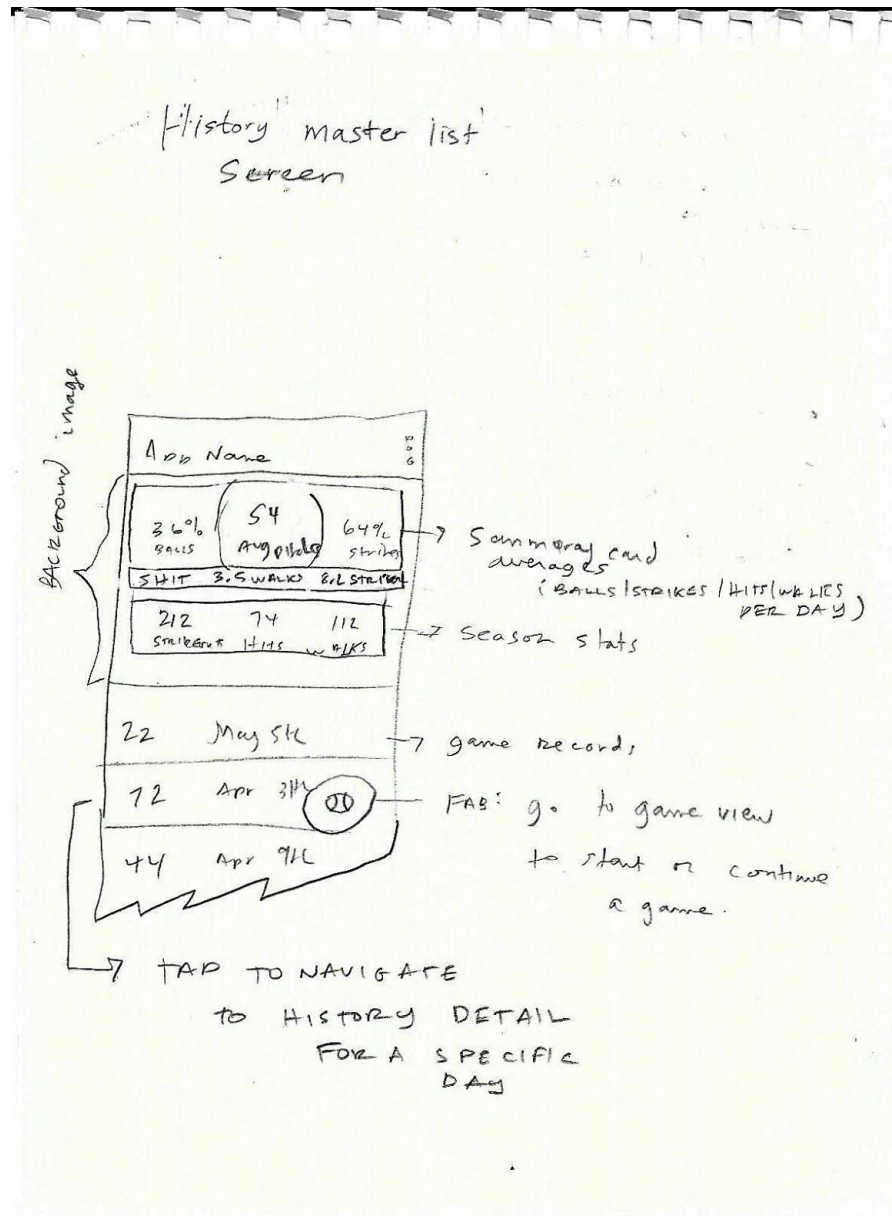
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

## Screen 1



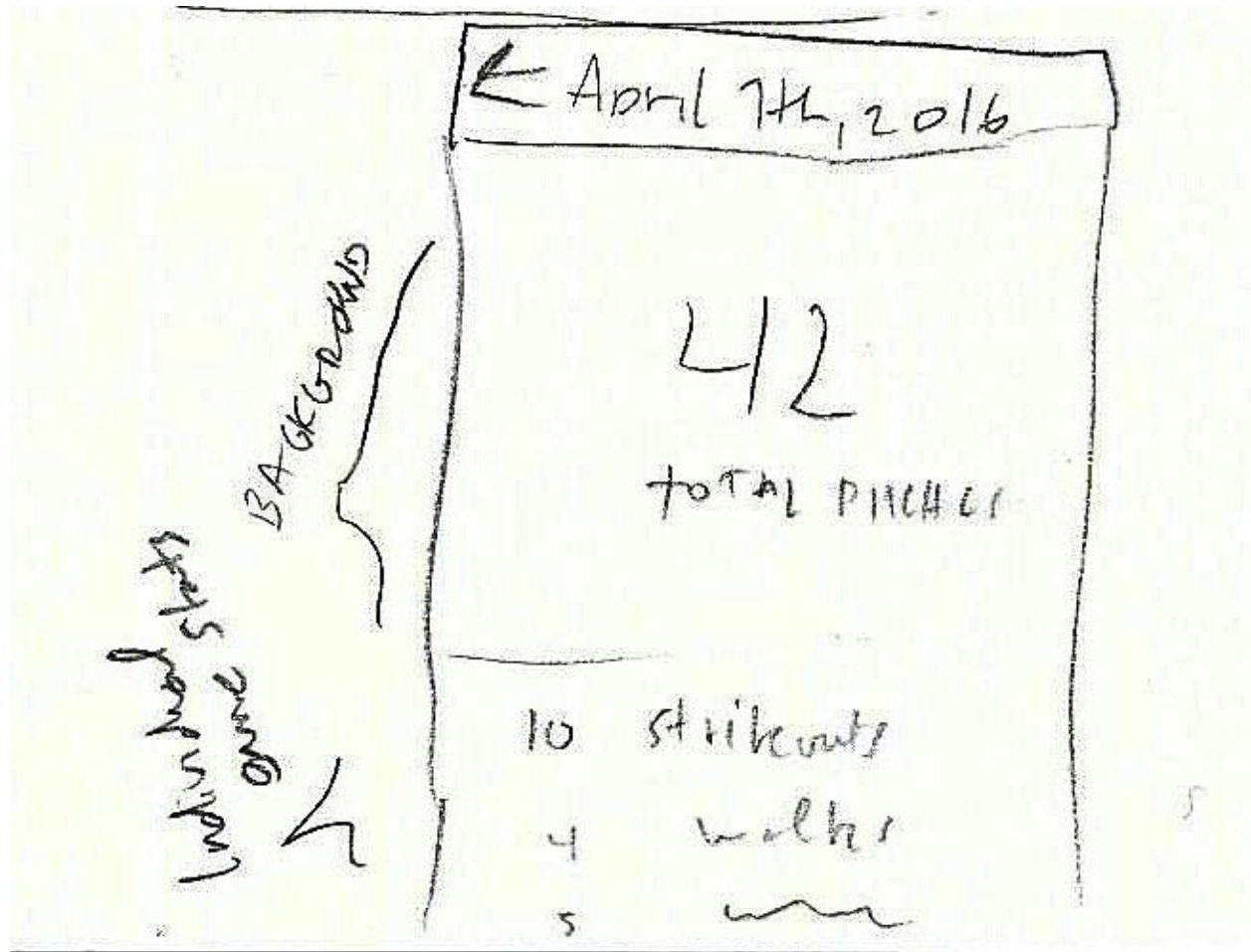
This is the game entry screen. This screen is all about entering the pitches for a current game. There are two primary action buttons, Ball and Strike. Tapping each button will update its respective label value and the total pitch count above. Surrounding the total pitch count is a circular graph that shows the ratio of balls to strikes. Below the total pitch count is the current date and a button for deriving your current location. Once your current location has been derived, it will appear in a user-friendly format below the date. In the app bar, there is an icon that will navigate you to the history master view screen.

## Screen 2



This is the history master view screen. It is all about giving you an overview of the player's cumulative pitching statistics. At the top of the list are two cards: one for statistical averages and one for cumulative totals (including balls, strike, hits, walks, strikeouts and total pitches). Below these two summary cards is the list of history items for games pitched in the past sorted chronologically by most recent game date. Each row is shown with a game's date and total pitch count. Tapping one of these history items will take you to the history item detail view screen. Anchored to the bottom right of the history master view screen is a Floating Action Button to return you to the game entry screen (either creating a new game or resuming the current game).

## Screen 3



This is the history detail view screen. It is all about showing you the details for a specific game. The app bar title shows the date the game occurred. The total pitch count is shown prominently atop a baseball-themed background image. Below the total pitch count are the individual statistics for strikeouts, walks, hits, balls and strikes. The location will be shown in the list, and any all-time highs for any of the statistics will be indicated when conditions are met. An activity transition will be made when navigating to and from this screen from the history master view screen.

## Screen 4

TABLET MASTER / DETAIL

AVERAGE CARD

TOTAL CARD

(24) MAY 5th

(72) Apr 30th

(54) Apr 14th

(10)

72

TOTAL PITCHES

April 30th

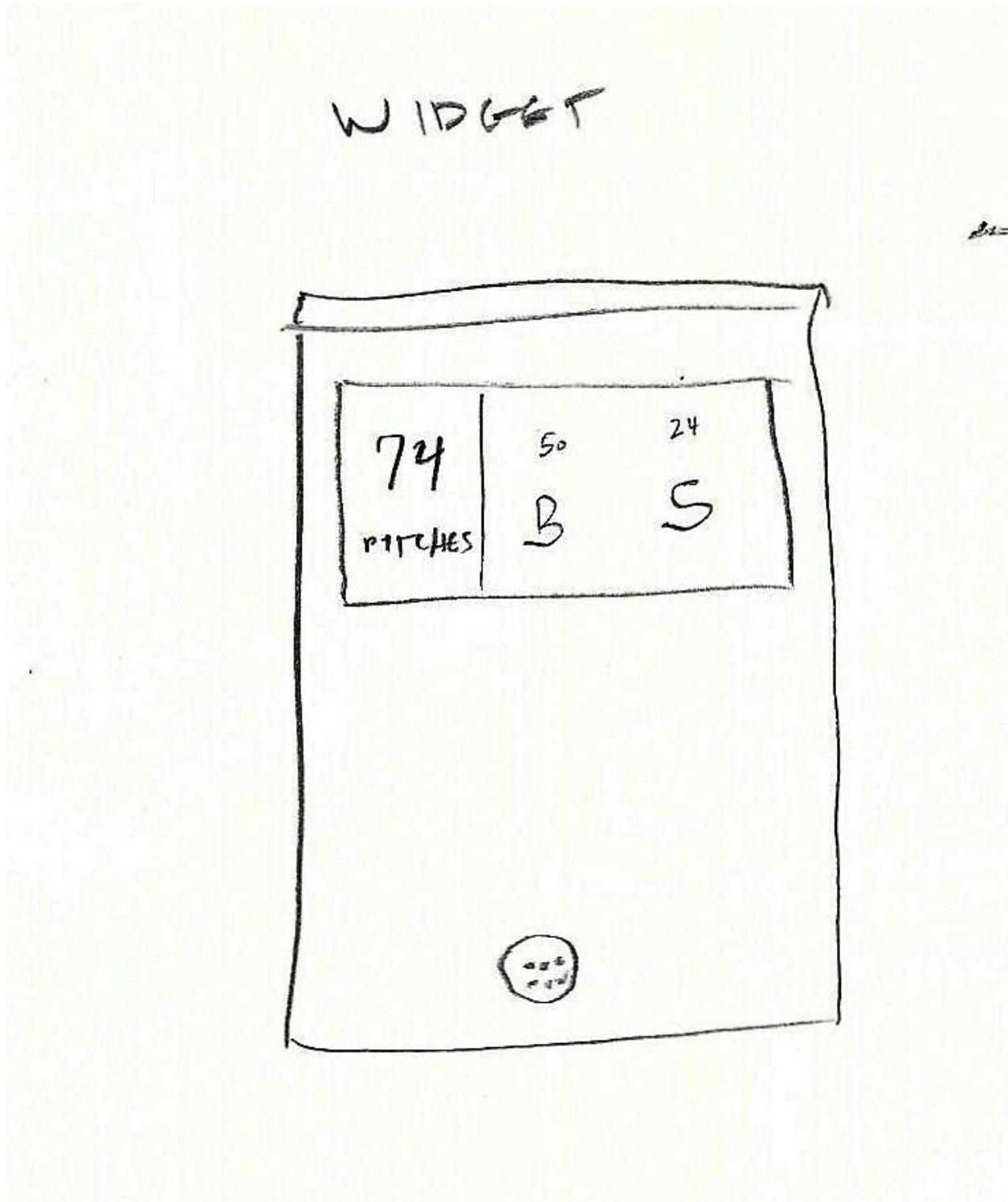
⑧ National Field Park

42	30	7	3	4
STRIKES	BALLS	STRIKES OUT	WALKS	HITS

This is the tablet master and detail view. When tapping a game item in the left panel list, the right panel will be populated by the full game details. A selection indicator will be shown for the currently selected game.



## Screen 5



This is the widget. It is a simplified way to enter balls and strikes. Tapping the total pitch count will navigate the user to the game entry screen within the app.

## Key Considerations

### How will your app handle data persistence?

Data for each game will be stored in a local SQLite database. Data will be inserted and read using a Content Provider.

### Describe any corner cases in the UX.

1. An empty state layout should be shown if the player does not have any previous games.
2. Location permission rationale should be presented to the user in a dialog.

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

1. Retrofit or Volley will be used to retrieve baseball-themed background image URLs from a remote content server.
2. Picasso will be used for caching and loading background images presented on the UI.
3. Sdkcheck will be used to check what version of Android the user is running in order to run OS-specific code like Activity transitions and Marshmallow permissions.
4. Butterknife will be used for simplifying the inflation of Views.
5. Otto or RxJava will be used to notify the UI that background images have been obtained from the remote server and are ready to be displayed.

## 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

- Create the base Android project in Android Studio and add to version control.
- Add external dependencies (libraries listed above and Google support libraries).

### Task 2: Implement UI for the Game Activity and Fragment



- Create phone and tablet variations of the layout in XML.
- Add event handlers and stub methods for updating the data set and displayed pitch counts on the screen.

### **Task 3: Implement the Game data model and data access**

- Create a SQLite helper class that interacts directly with the SQLite database. This includes the ability to create tables, insert and update a game record, and retrieve current game statistics.
- Implement a Content Provider that accesses database queries. This will be how the UI interacts with the data model.
- Implement Loader calls that use the above-mentioned Content Provider to update the UI.

### **Task 4: Implement the location-tagging feature**

- Add a button that will attempt to retrieve a human-readable location to tag the game with. While obtaining the location, update the UI to show progress. If a location is obtained, update the UI.
- Add Android Marshmallow permission checking for the location permission. This includes handling cases where the user denies the permission.

### **Task 5: Implement the History master view and data population**

View-specific tasks:

- Create RecyclerView Adapter and associated View Holders for the history Fragment.
- If there aren't any games in the history, display an empty state message.

Data-specific tasks:

- Populate a dataset for the history list by first writing a SQL query to retrieve all games sorted by recency.
- Provide the ability to access this data via a Content Provider.
- Implement Loader call to update the UI with the dataset.

### **Task 6: Implement the History detail view**

- Create a detail screen that shows the total number of pitches, number of balls, strikes, strikeouts, hits, walks, and tagged location (when data is present).

- Implement an Activity transition when navigating to this page from the master view.
- Implement a parallax scrolling effect when scrolling this page's content.

### Task 7: Create a widget for entering balls and strikes, displaying totals

- Create the Remote Views for gathering and displaying the current pitch, ball and strike counts.
- Wire up the aforementioned Content Provider to the update the widget UI based on the current data.

### Task 8: Retrieve and present baseball-themed background images

- Initiate network calls to retrieve image URLs.
- For a given URL, load the image and present it as the background image on the Game entry screen.
- If images cannot be obtained, ensure that the default background appears.

### Task 9: Implement analytics

- Configure server side Google Analytics for this app.
- Track page view events.

---

#### Submission Instructions

1. After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone\_Stage1.pdf**"