GitHub Username: smeschke

Big Top Tricks

Description

As you stand there with three balls in your right hand and two in your left, you wonder, “How am I ever going to juggle five balls?” Others make 5, or even 7 ball juggling look so easy!

This app breaks down what seems like an impossible task into manageable steps Highly organized steps will give the user a good idea of learning rate and progress toward the goal.

The steps will include a combination of video and text based instruction. Some of the steps will have an activity for the student to complete.

As users progress though the app, things like time trained and records achieved are logged into a database. The data will be visualized in graphs and will be easily shareable to one of the juggling social networks.

Intended User

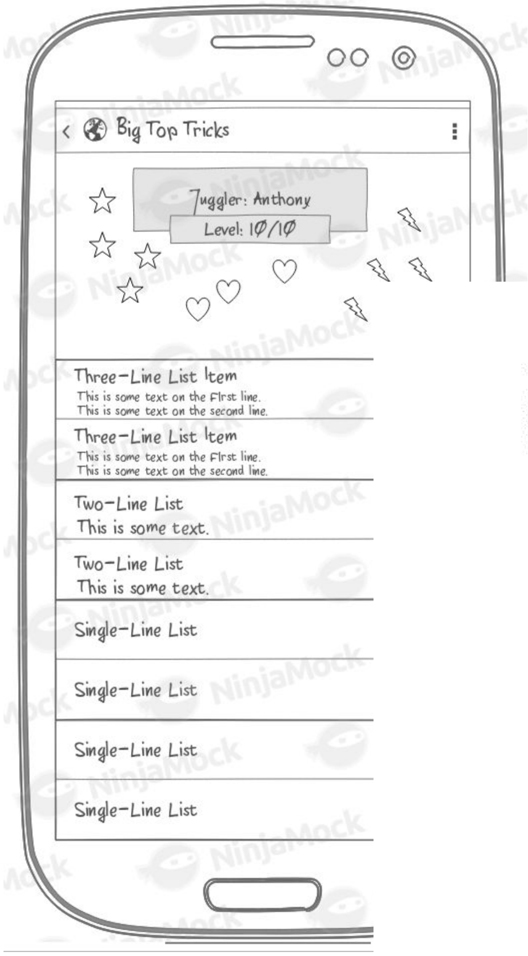
The intended user is anyone that is interested in learning juggling.

Features

- Video and text based instructions
- Visualization of training data
- Ability to record information about training sessions
    - Record time trained and personal record (# catches or time)
    - Record short video clips

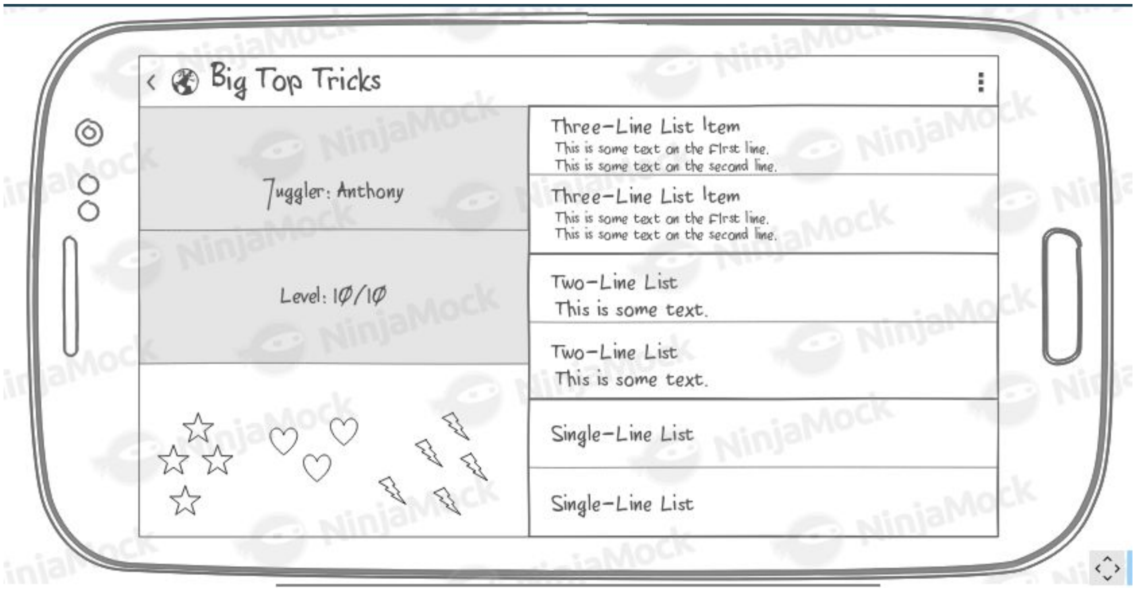
User Interface Mocks

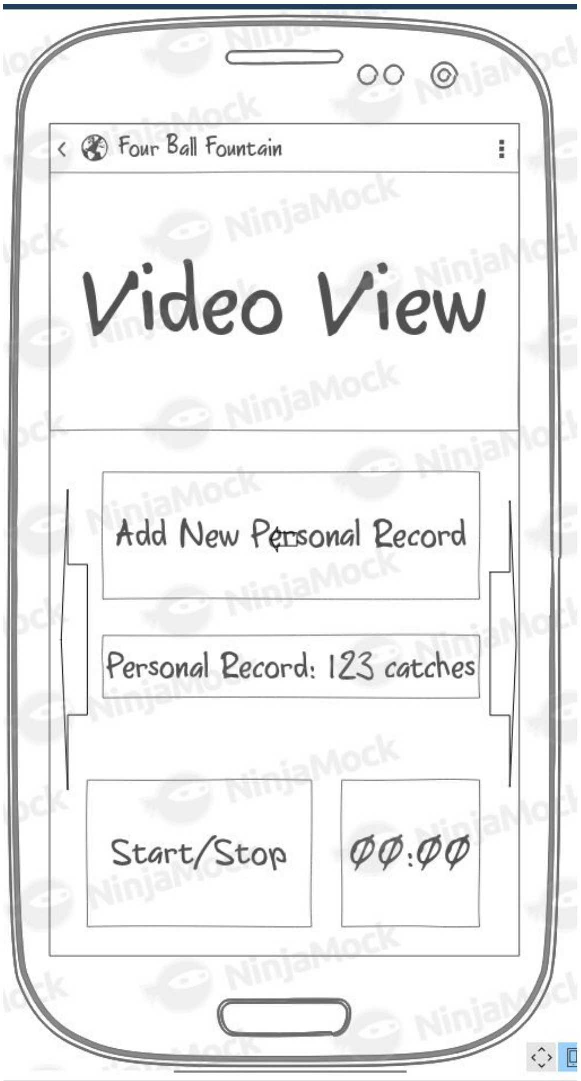
https://ninjamock.com/s/2V7ZFTx

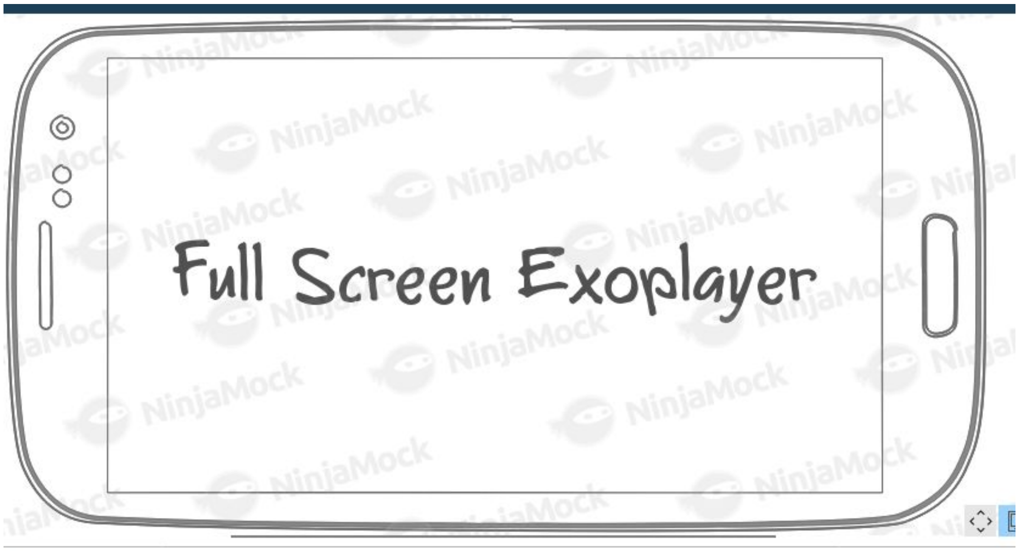


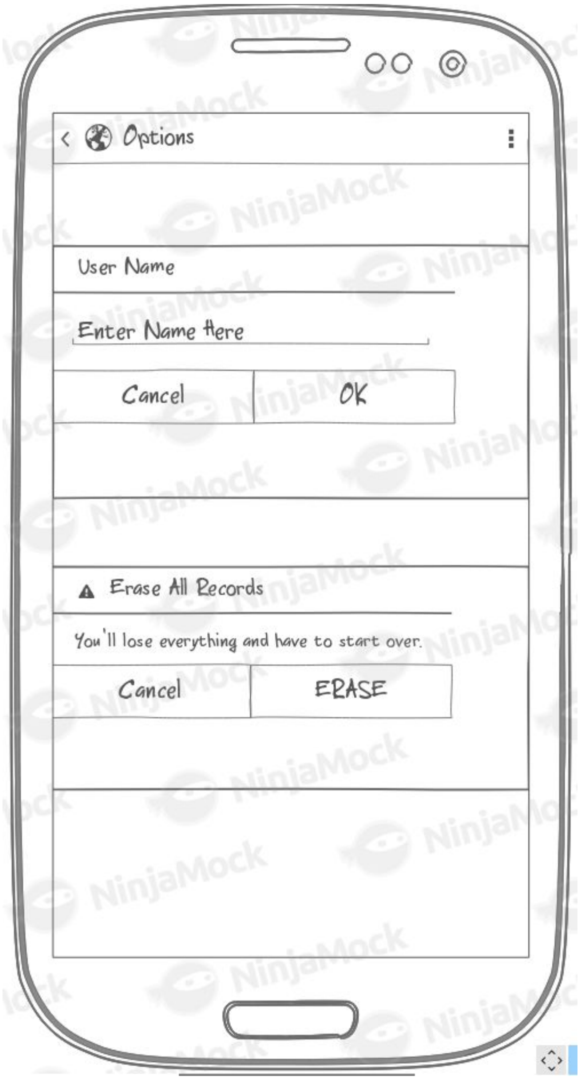
These mocks show the Main Activity. The app will launch to this screen, where the user can get a snapshot of thier progress, and naviage to the various tricks.

This app does not use the master-detail flow, but when the device is used in landscape orientation, the jugglers snapshot of progress is shown in the left pane, along with the list of tricks in the right pane.

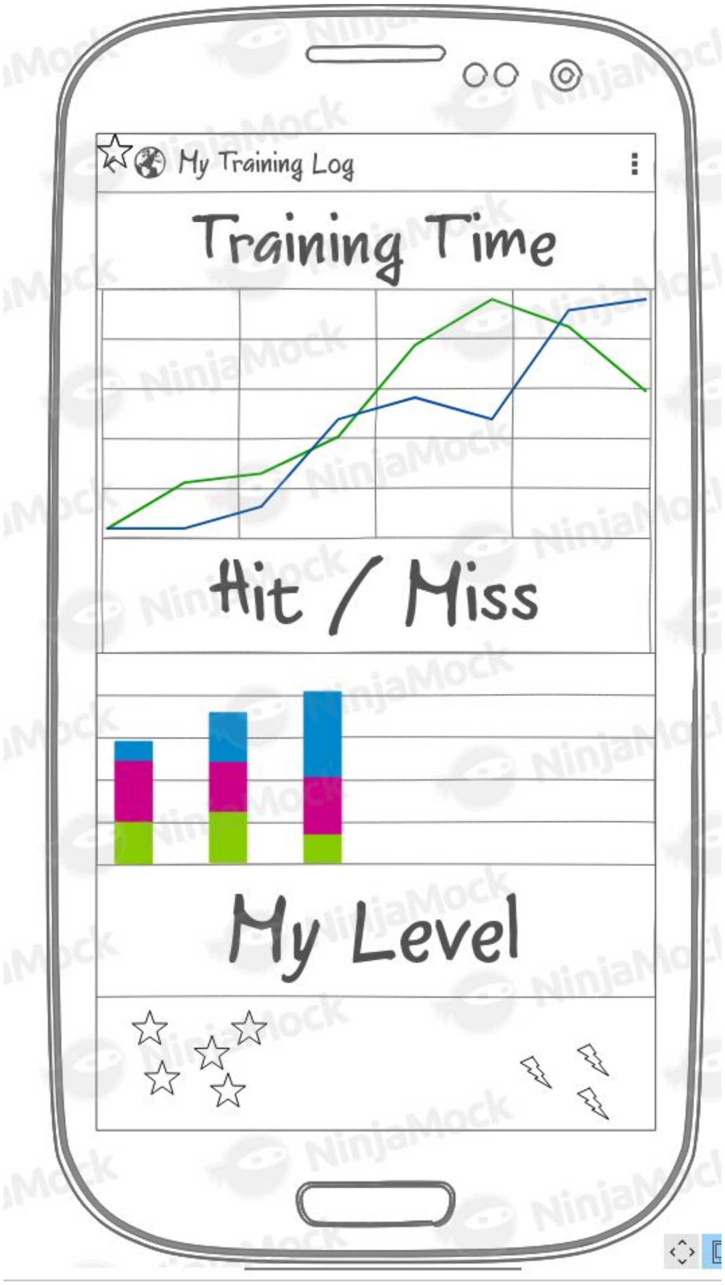



Four Ball Fountain
Video View
Add New Personal Record
Personal Record: 123 catches
Start/Stop
ØØ:ØØ


Full Screen Exoplayer



You'll lose everything and have to start over.



Key Considerations

How will your app handle data persistence?

Data persistence is a key feature of this app, and is used to track the users progress and learning rate. The data will be stored in an SQLite database and accessed using a content provider and cursorloader.

Describe any edge or corner cases in the UX.

It is likely that the user will briefly navigate away from the app to reply to a message or answer a call. There will be lots of edge cases involving videos. Frequently, users will watch part of a video and then navigate away. The video should preserve the position so that the user does not have to skip through the video.

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

Exoplayer will be used to display videos.

A graphing library will be used to display the user's data.

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

JTV.com will be used as an endpoint for the videos.

Next Steps: Required Tasks

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.

A separate Android studio project will be created to demonstrate each of the following features:

- Graphing Libraries
- Playing videos with Exoplayer
- Uploading users training data

Task 2: Implement UI for Each Activity and Fragment

Start with smallest screens first. This app will be designed for left handed users.

Build the UI for the main activity (list of tricks):

- Item for the image at the top of the list
- Item for each element (trick) in the list

Build the UI for the detail activity (detail of one trick):

- Navigate to the previous or next step
- Add elements (Exoplayer view, text view)

Build the UI for the digital badge activity:

- This activity is static, but requires a lot of text view filling in
- A screenshot of this activity will be shared, so it's extremely important that this activity looks good on all devices.

Task 3: Implement a database

The database is the heart of this app. After the UI is finished, a database can be added. The following tasks will wire the database to the app's UI:

- Create database
- In the detail activity: Wire time trained and PR to the database
- In the digital badge activity: Wire the results to the graph

Task 4: Graph the data.

Graphing the data is something that is new to me, and I will have to learn how to use a graphing library for this purpose. The following graphs need to be created to output the data:

- Time trained graph (line graph)
- % hit. Is a trick ready for performance? (bar graph)
- Tricks trained (exploded pi graph)

Task 5: Test!

Testing is required to ensure that the database functions properly and the app has a pleasant appearance on all types of devices. The following test need to be run:

- Can the user enter a value that crashes the app?
- Is the Exoplayer being released and initialized correctly?
- Does the navigation work properly.

Task 6: Upload Assets

This app should look great on high resolution devices, as well as older devices with smaller screens.

- Use vector graphics (Inkscape)
- Get translations