Contents

Description........ 2

Intended User........ 2

Features........ 2

User Interface Mocks........ 2

Screen 1........ 3

Screen 2........ 4

Screen 3........ 5

Screen 4........ 6

Screen 5........ 7

How will your app handle data persistence?........ 8

Describe any edge or corner cases in the UX......... 8

Describe any libraries you'll be using and share your reasoning for including them......... 8

The application will be written solely in the Java Programming Language........ 8

Describe how you will implement Google Play Services or other external services......... 9

Next Steps: Required Tasks........ 9

Task 1: Project Setup........ 9

Task 2: Implement UI for Each Activity and Fragment........ 9

Task 3: Implement data storage......... 10

Task 4: Error Handling........ 10

Task 5: Implement widget........ 10

GitHub Username: MahmoudMB

MyTasks

Description

The application help people to capture their to do lists and tasks and share it with friends.

Intended User

The App is for anyone who wants to remember his tasks and share a list of task with a friend.

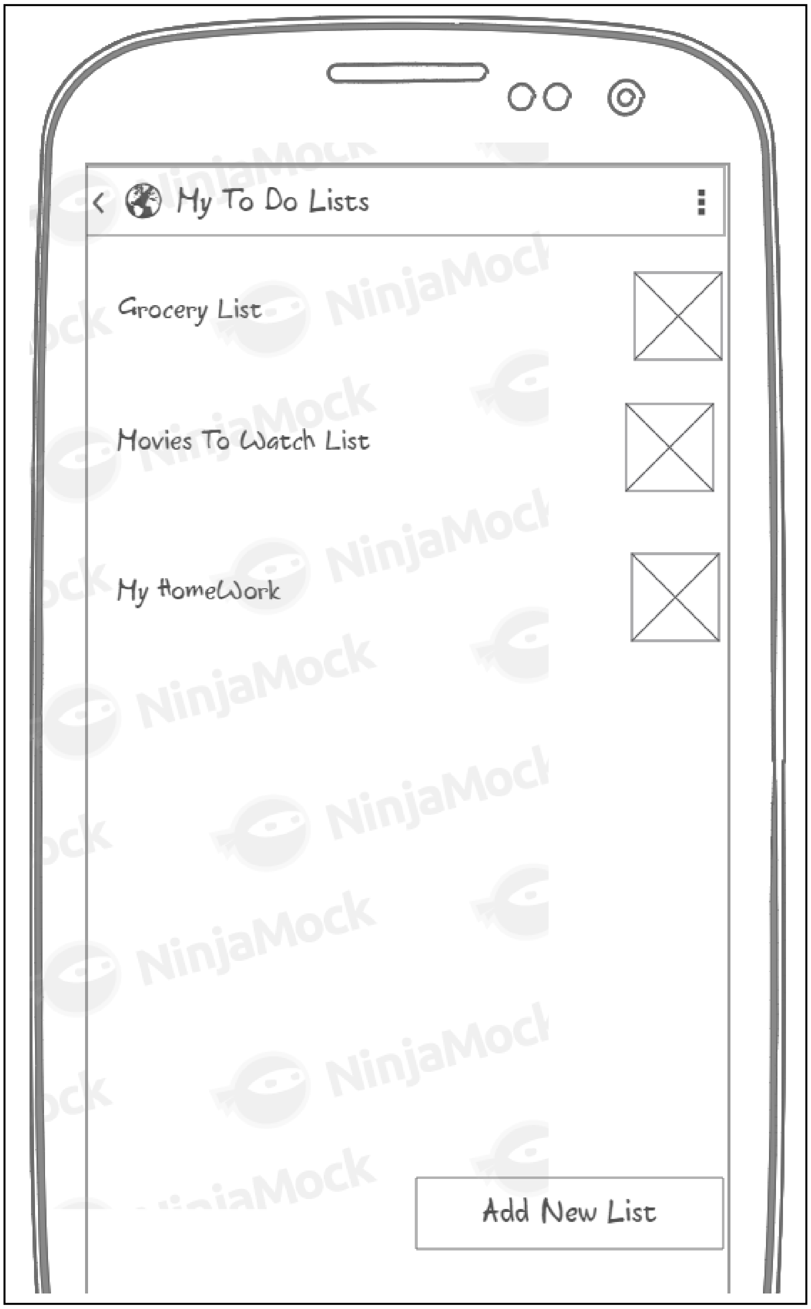
Features

- Create a to do list.
- Share a to Do list with a friend
- Display the friends tasks
- Assign a task to a friend

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.

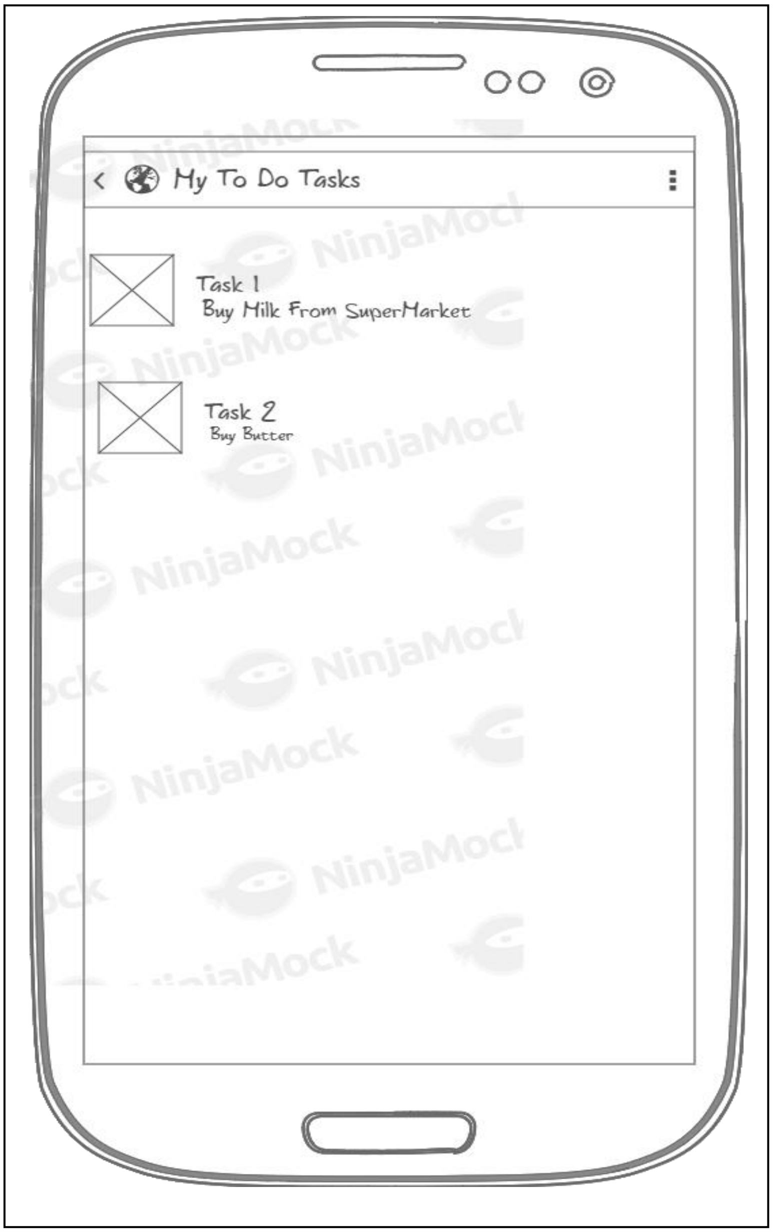
Screen 1



Description:

in this screen, it contains a listview containg the lists of tasks, the user will be able to see his lists of tasks and will be able to add a new list of tasks.

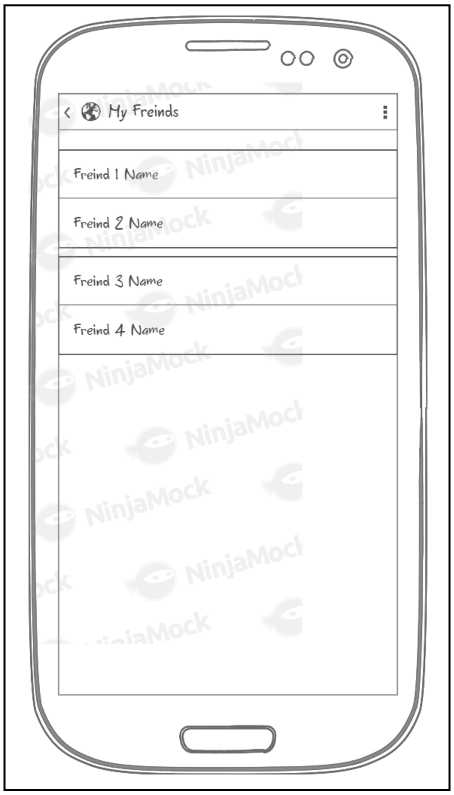
Screen 2 (The tasks List)



Description:

in this screen the user will be able to see the tasks of each list he create.

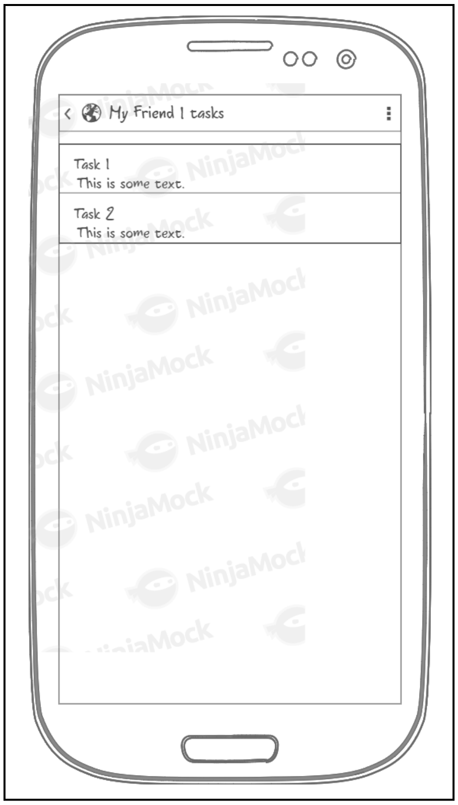
Screen 3 (The Friend List)



Description:

in this screen the user will be able to see a list of his friends.

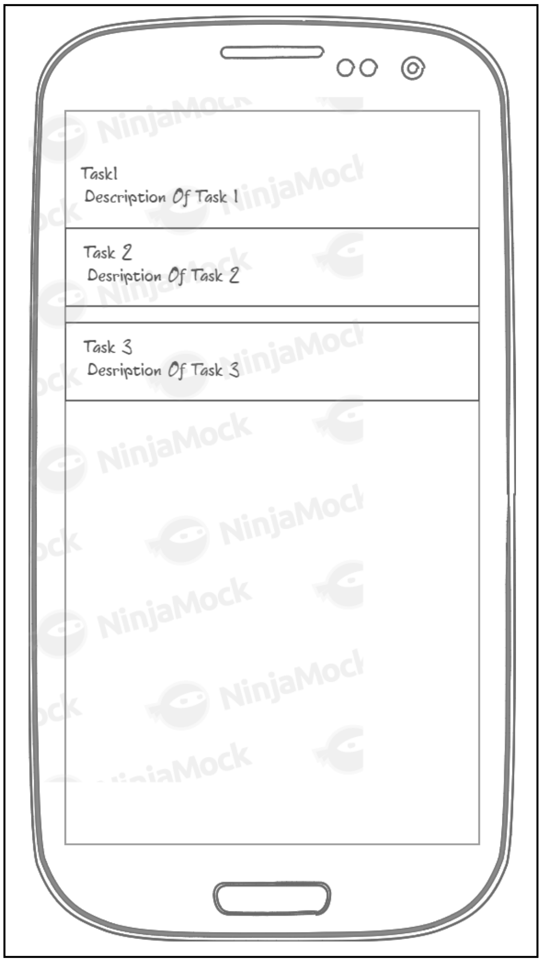
Screen 4 (The Friend Shared Tasks)



Description:

If the friend has a shared tasks with the user, these tasks will appear in this screen.

Screen 5: (The App Widget)



Description:

The Widget will allow the user to see his list of tasks.

Key Considerations

How will your app handle data persistence?

The Data will be stored in Firebase Realtime Database.

Describe any edge or corner cases in the UX.

- **Unstable or missed network connection**:
  the application must not crash in that cases and show a message to the user.

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

The application will be written solely in the Java Programming Language

- butterknife for binding android views. (butterknife:8.8.1)
- firebase-database:16.0.1
- firebase-storage:16.0.1
- firebase-auth:16.0.1
- firebase-core:16.0.1.

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

- Firebase for Realtime Database(firebase-database:16.0.1).
- Firebase Analytics (firebase-core:16.0.1)

The application will use Firebase Realtime Database and Analytics which both depend on Google Play Service

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 new project in Android Studio.
- Add the Libraries Dependencies in gradle.
- Design the Structure for the Database.
- Configure the AndroidMainifest.xml to enable various permissions necessary for accessing the internet.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for User Login
- Build UI for List of To do tasks
- Build UI for List Of Friends
- Build UI for my friends Tasks.

Task 3: Implement data storage.

We will store the data in a Firebase database.

Task 4: Error Handling

Find And handle errors that might happen in the application like:

- When there is no access to the internet.
- When there is no data returning from Firebase database.

Task 5: Implement widget

- Create layout
- Get the user tasks
- Display it on the widget
- The Widget will be updated using IntentService when a new task is added to list of tasks.
- The Database Will be updated using IntentService, if the user Marked a task as completed on the widget.

Task 6: Polish App By Implementing Material Design Principles

- Pick a color scheme with primary and accent colors.

- Make sure text and images have enough space around the and are aligned along keylines.
- Make simple transitions between activities.

Task 7: Allowing for Localization and Making the App Accessible

- Enables RTL layout switching on all layouts
- App is written solely in the Java Programming Language.
- App keeps all strings in a strings.xml file
- Ensure the app can be navigated using a D-pad
- Provide content descriptions for all images.

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**”