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: MahmoudAhmedElshahat

Sense

Description

Sense will make users connect to each other in real time chat application

Intended User

For everyone who have friends or family to communicate with them in text chat way

Features

- Send Messages
- Accept or refuse friends requests
- Find other people that use application and add them to friend requests

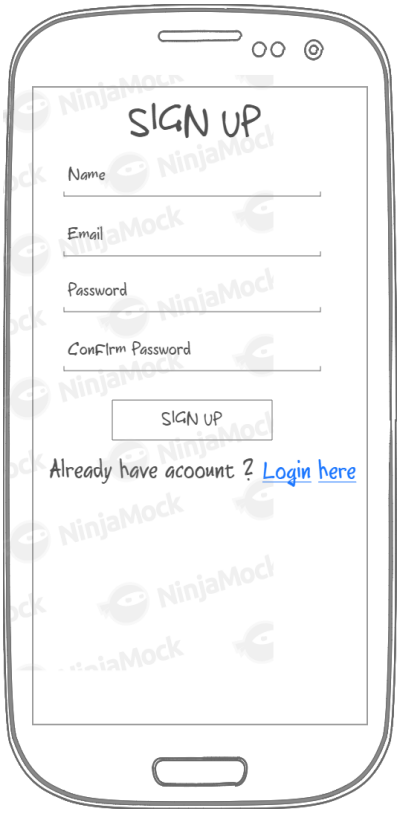
User Interface Mocks

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.

Login Screen (First Screen Launched)

The first screen in app user can login with username and password that he is created , or even by twitter , gmail or facebook

Register Screen



Here user can register an account just by add email and password and his name

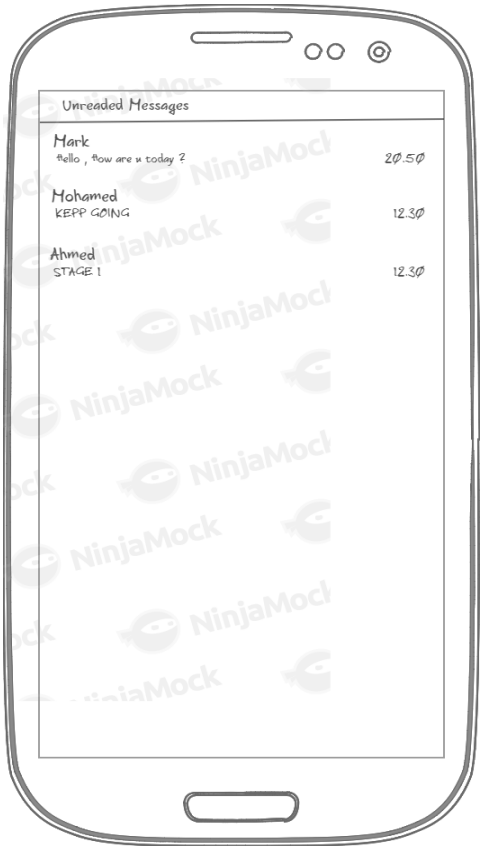
Main Screen

Here is a list of friends that he can contact with them , and tap layout for navigation between Chat and Requests from others to add you , or even Find tap that you can search to find friend or by exploring applications users

Chat Room

Here user can type message in realtime way with his friend

App widget



Widget for showing nuread message in Sense application

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

Using Firebase realtime database data will be fully handled in real time way and Content Provider library for offline checking message will be used too

Describe any edge or corner cases in the UX.

- Incorrect passwords-emails in login : will be get a message that this data is not right

- Network unavailability : User can access all messages that stores offline and when network returned all new messages will added
-

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

Glide : for user image uploading and showing
Realm : for storing and retrieving messages

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

Sign in with google : Login with email or password or even google gmail account
Google ads : adding ads to main screen
Firebase : Analytical , authentication , notification and realttime database

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

- Add all wanted dependencies in gradle file
- Building activities xml
- Start step by step from register to login then to make a chat

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for Login System (Login – register - forgot)
- Build UI for MainActivity
- Build UI for each fragment in main activity

Task 3: Implement Logic for application

- Using Firebase authentication system
- Using Firebase realtime database for messaging
- Using Content provider to save data and loader to load it

Task 4: Main points

- Widget for a single user chat
- Add adMob to MainActivity
- Bonus : Add features like (change background screen – change chat colors .. etc)

Task 5: Notifications Handling

- Backend development that make push notification easier
- Front end in android by AsynTask that handle http get and post method for push notification
- Each notification has a replay button and opens to target chat correctly

Add as many tasks as you need to complete your app.

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**"