Storyteller

Description

The goal of this app is to connect people across the globe who like to read or write stories. Some of the different genres will be presented and users can choose in which one their story belongs, or what kind of story they want to read.
Users will be able to save their favorite authors and stories and access them anytime.

Intended User

Everyone is a storyteller. App is mainly for people who like to write or improve their writing skills, but it's also for those who like reading.

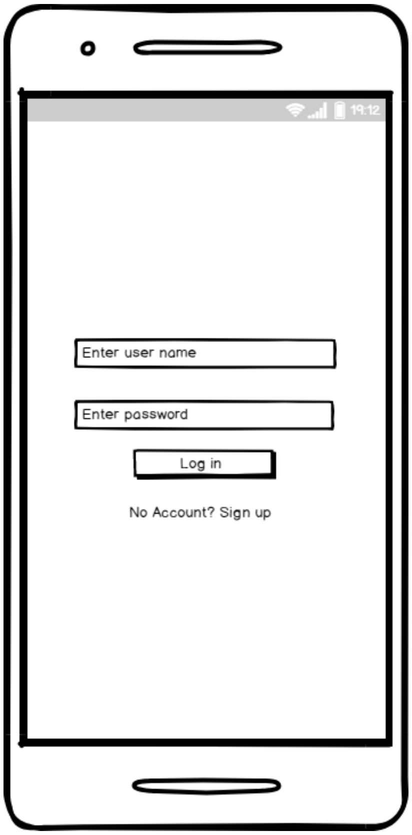
Features

List the main features of your app:

- Creates user accounts
- Stores all of the stories in the database
- Allows users to read and write stories

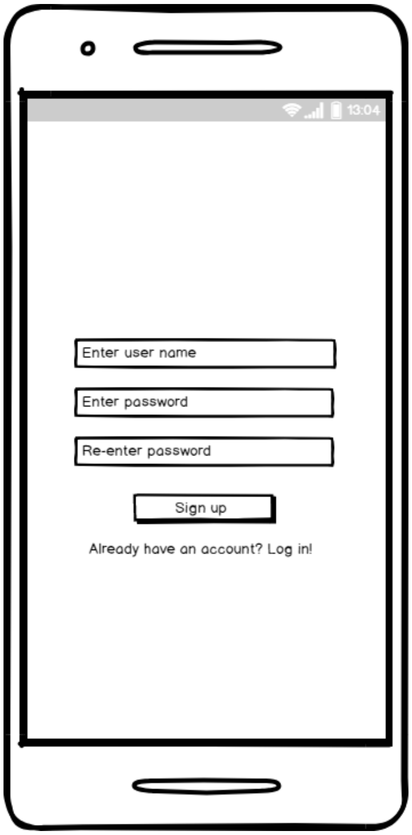
User Interface Mocks

Screen 1



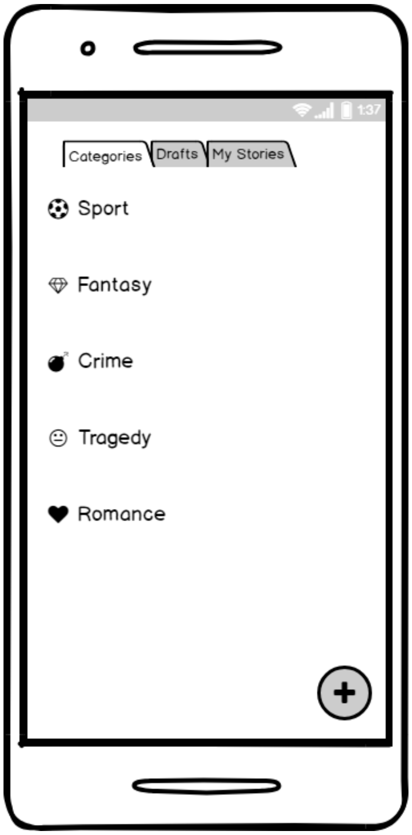
Login screen where user can log in using his password and email if he has an account. If he’s not registered yet, user can click on “No Account? Sign up!” and create an account.

Screen 2



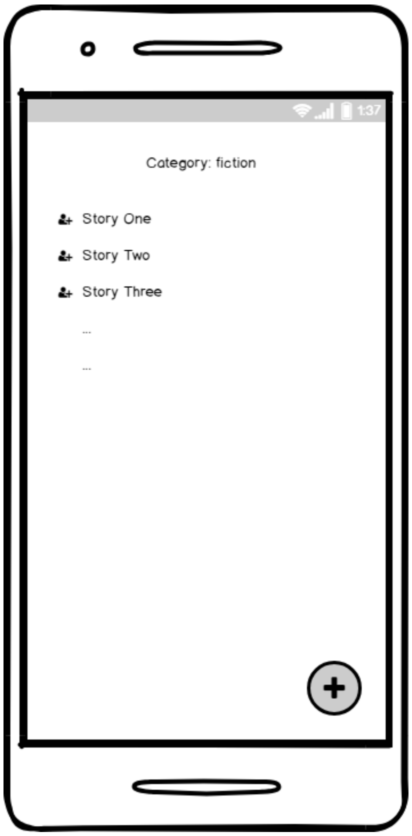
Sign up screen where users can create their account.

Screen 3



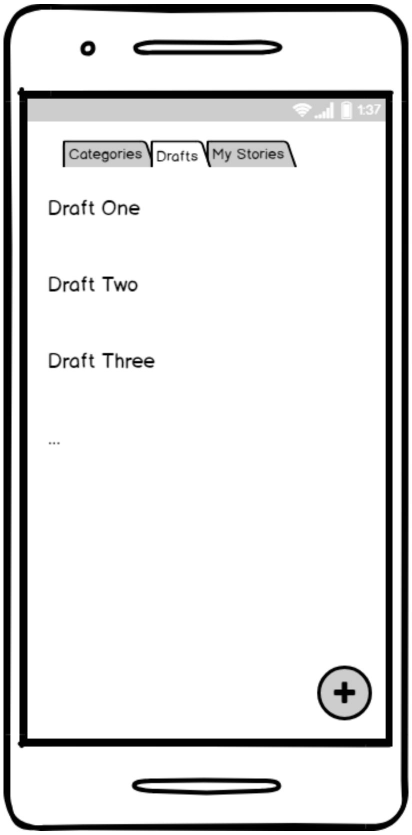
Main activity screen. In first tab user will be able to go through categories and read stories he is interested in. Users can also click on the ‘+’ fab and create their own stories.

Screen 4



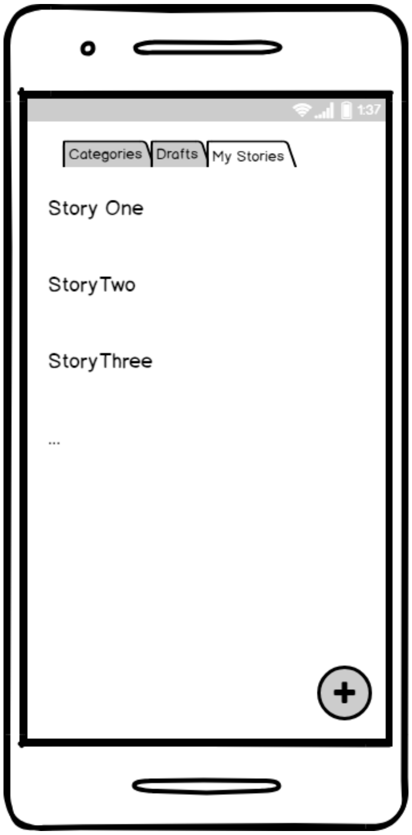
Once category is selected, list of stories in that category will be displayed. Clicking fab will add a new story in selected category.

Screen 5



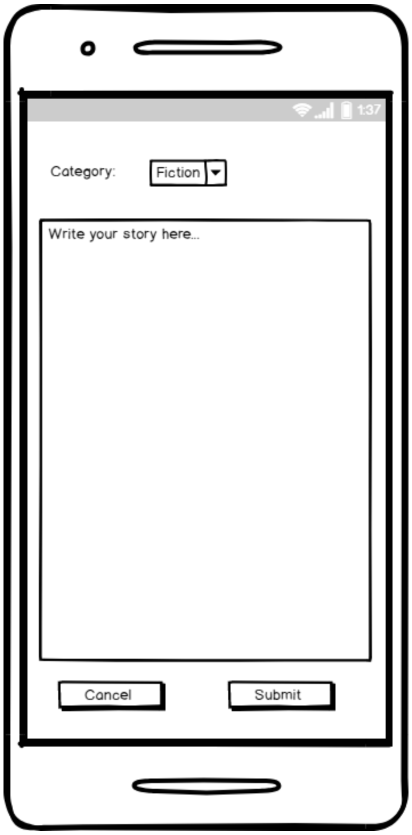
Drafts screen displays all the stories user has written, but hasn’t published yet.

Screen 6



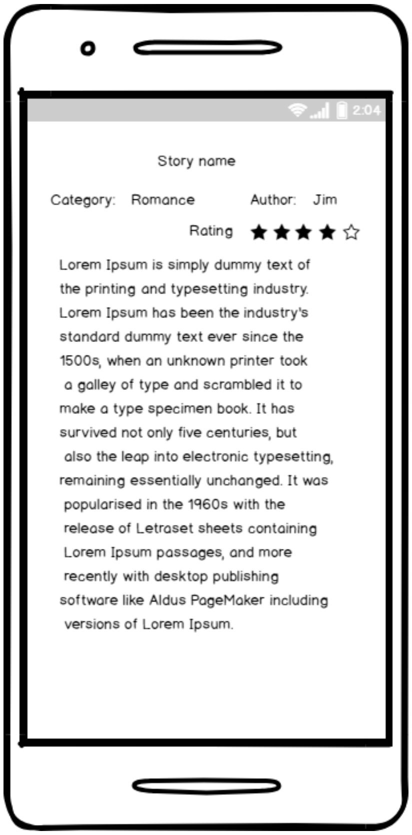
My Stories tab is an activity with all of the stories user has written and published.

Screen 7



Clicking on the ‘+’ fab will take user to an activity screen where he can write his stories. Clicking back button or exiting application in any other way will save the text as a draft, canceling will delete it, and clicking submit button will publish their story.

Screen 8



Story name

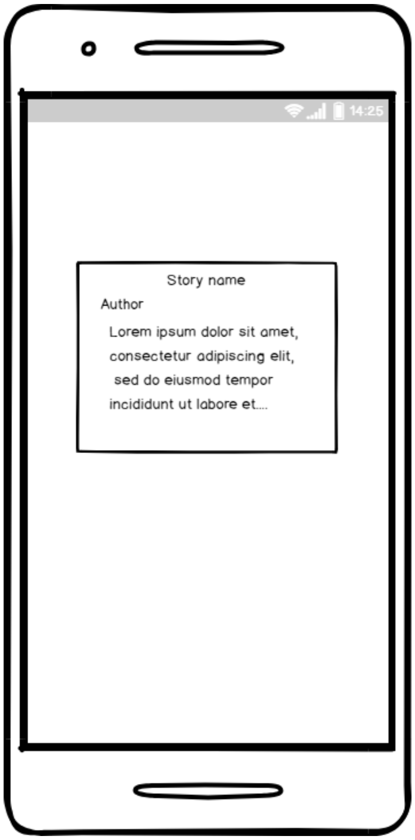
Category: Romance Author: Jim

Rating ★★★★☆

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

A screen where user can read and rate story.

Screen 9



Story name
Author
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor
incididunt ut labore et....

App widget will display a preview of newly added stories.

Key Considerations

How will your app handle data persistence?

Data will be stored using Firebase Realtime Database.

Describe any edge or corner cases in the UX.

Fragment will be popped from stack on back navigation.

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

Gson – will be used to convert Java objects into their JSON representation, and the other way around.

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

Firebase Authentication – used to create user accounts. Users will be able to sign up using their email and password. This is so they can log in on any device, and it is necessary to create a user account since we're using the database.
Firebase Realtime Database – used to store/retrieve data about users and stories.

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 project in Android Studio
- Create theme for project
- Configure styles, dimensions, colors and strings
- Create a folder structure
- Add dependencies
- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Login and Signup screens
- Build UI for main fragment that will include three tabs – story categories, drafts and user's stories
- Build UI for fragment displaying stories in certain category
- Build UI for story fragment (reading)
- Build UI for story fragment (writing)

Task 3: Add Firebase Authentication

- Use Firebase Assistant to add Firebase to the project
- Go to Firebase console and add a new project
- Enable user authentication

Task 4: Implement Firebase Authentication

- Create sign up and log in screens
- Add required functionality so users could log in if they have an account or sign up if they don’t have one yet

Task 5: Add Firebase Database

- Add Firebase Database to the project using Firebase Assistant (check if newest dependencies have been added)
- Go to Firebase console, enable database and set up rules for it

Task 6: Implement split action bar

- Create layout for main screen
- Add functionality to fab
- Create recycler view list for each of the screens

Task 7: Complete story reading/writing fragments

- Create layout for story reading/writing fragments
- Store all of the user stories in the database
- Retrieve stories from database and display them to the user

Task 8: Add a widget

- Create widget layout
- Display preview of newest stories using widget

Technical aspects:

1. Programming language

App is written solely in the Java Programming Language

2. Library versions

App uses Gradle version 4.4 and Android Studio version 3.1.3

Dependencies:

```
implementation fileTree(dir: 'libs', include: ['*.jar'])
implementation 'com.android.support:appcompat-v7:27.1.1'

implementation 'com.android.support.constraint:constraint-layout:1.1.2'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'com.android.support.test:runner:1.0.2'
androidTestImplementation 'com.android.support.test.espresso:espresso-
core:3.0.2'

implementation 'com.google.firebase:firebase-auth:16.0.2'
implementation 'com.google.firebase:firebase-core:16.0.1'
implementation 'com.google.firebase:firebase-database:16.0.1'

implementation 'com.squareup.retrofit2:converter-gson:2.3.0'
implementation 'com.android.support:cardview-v7:27.1.1'
implementation 'com.android.support:recyclerview-v7:27.1.1'
```

3. Accessibility support

App will be written following the basic accessibility principles including content labeling, touch target size, color contrast, and view attributes.

4. RTL support

Right to left support will be enabled.

5. Data persistence

App will store all of its data in Firebase Realtime Database and will access it using a JobDispatcher.