**Capstone_Stage1**

---



## This app is written solely in the Java Programming Language

**GitHub Username: https://github.com/AuraLoredana**

**App NAME: TCB (abbreviation from Taxiul cu Bomboane)**

**Description**

TCB it is a mobile app for a Nonprofit organization, that helps children from hospitals, by attracting donations and/or volunteers. The app has also a chat, implemented with Firebase, in order to help volunteers to socialize. The purpose of this application is to help children from the Romanian hospitals and convince people to join the organization and to make a change in the Health System of Romania.

**Who is your intended user?**

This app it si for everyone that wants to help, whether it is through a donation, or join us as a volunteer in hospitals.

- Features
- Saves information
- Chat
- Messages
- E-mail
- Login
- Registration
- Share
- Rating

**User Interface Mocks**

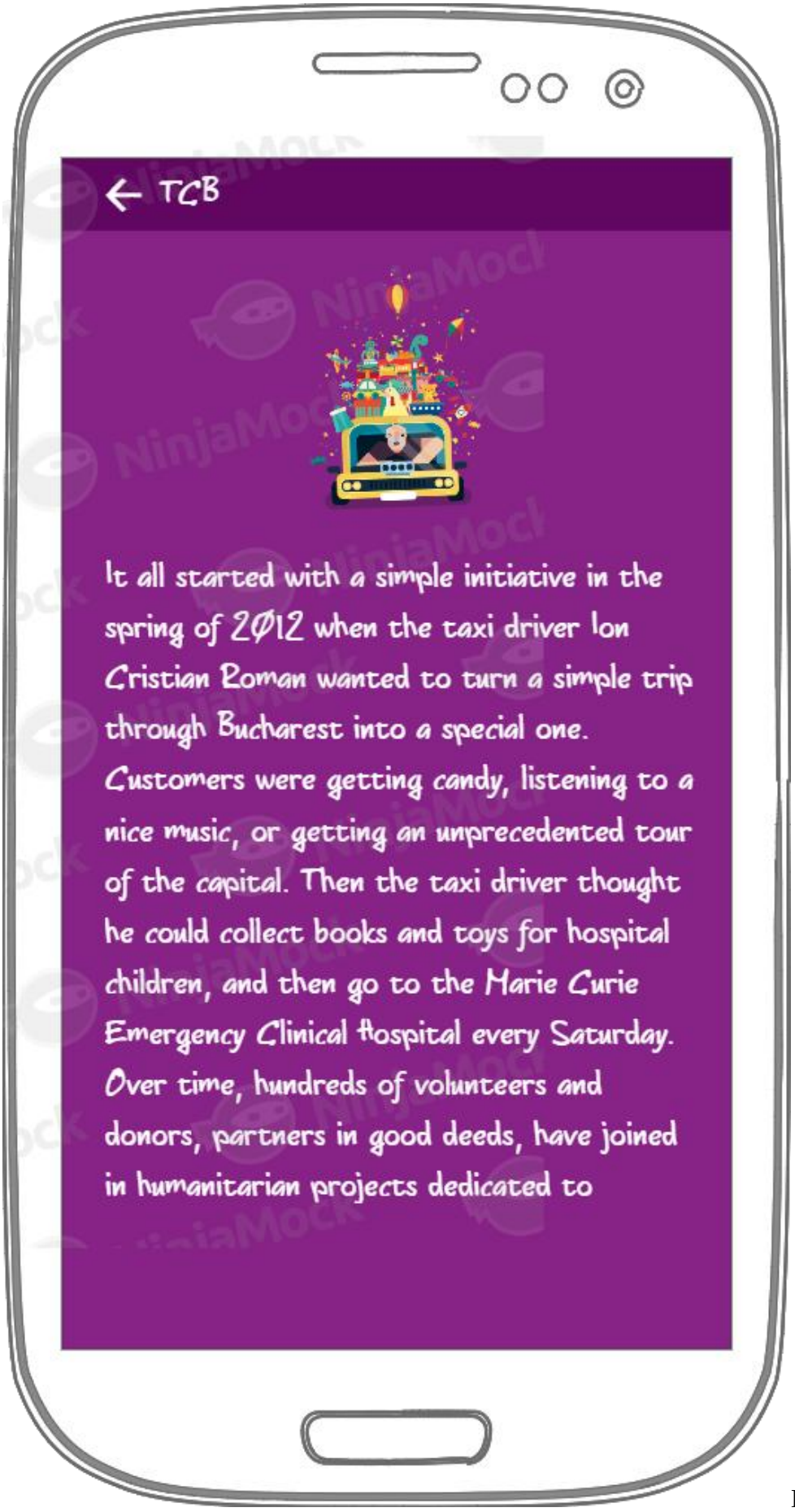
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1


TCB
About
Our Projects
Become a Volunteer
Donate
Chat
Contact
Rate us
Share

Menu on the

left: Navigation Drawer Menu




Description about the company “Taxiul cu Bomboane”. I have implemented a parallax efect.


Chat box


Registration Screen


A donation page

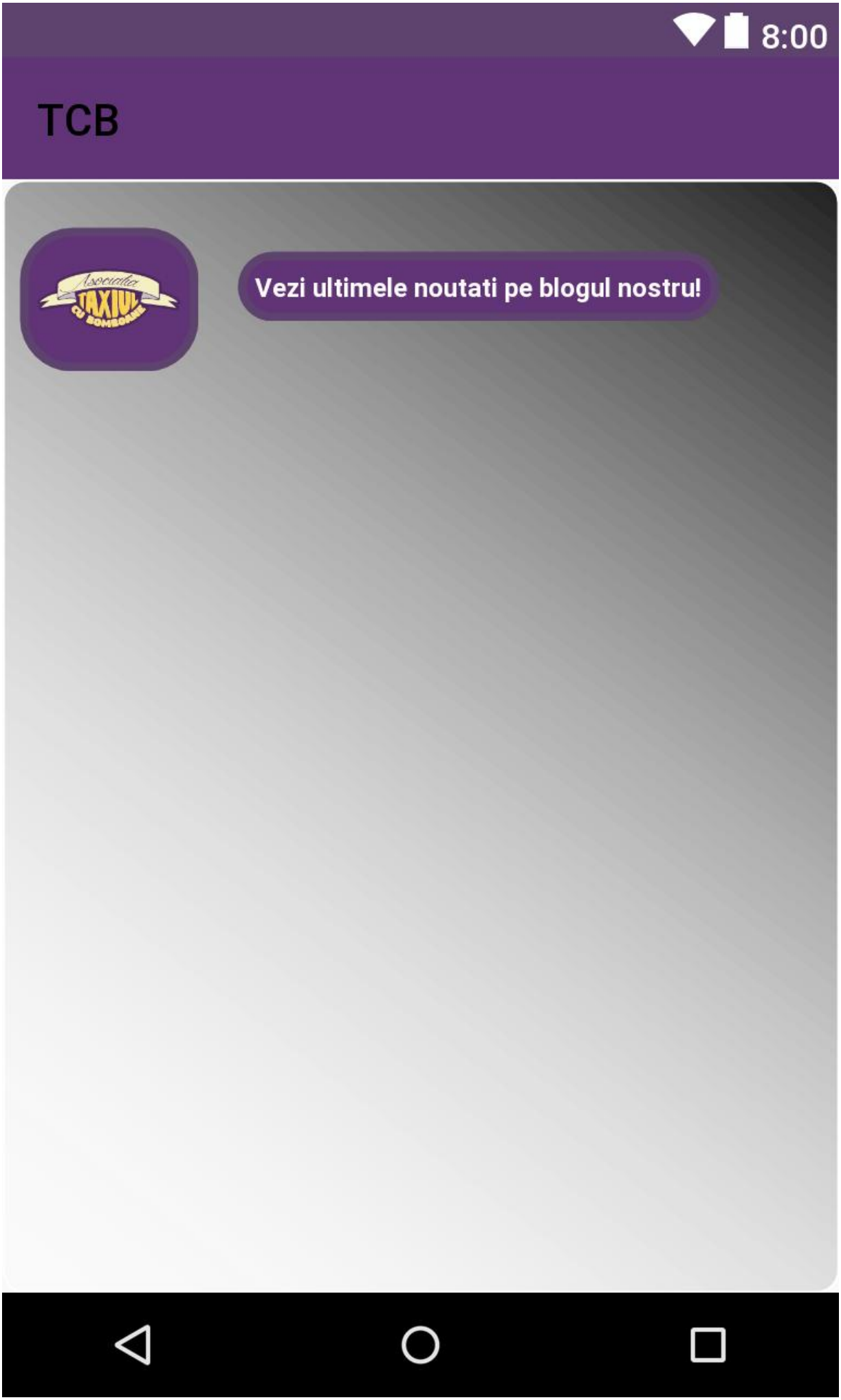

8:00
TCB
Vezi ultimele noutati pe blogul nostru!


*Figure Widget1*

**UI mock for app's widget- home screen widget: when you click on the button it opens an URL for the blog page of the organization**


**UX case if no data/internet is there**

**Key Considerations**

**How will your app handle data persistence?**

**Describe how your app with handle data. (For example, will you build a Content Provider or use Firebase Realtime Database?)**

I have used a Firebase Realtime Database for the implementation of the chat, login and registration of the user.

**For example, how does the user return to a Now Playing screen in a media player if they hit the back button?**

To return to a previous page, just press the back button on their phone.

**Describe any libraries you'll be using and share your reasoning for including them.**
I have used the following libraries:

**implementation 'com.google.firebase:firebase-core:15.0.2'**
**implementation 'com.firebase:firebase-client-android:2.5.2+'**
**implementation 'com.firebaseui:firebase-ui:0.6.2'**
**implementation 'com.google.firebase:firebase-storage:15.0.2'**
**implementation 'com.google.firebase:firebase-database:15.0.0'**
**implementation 'com.google.firebase:firebase-auth:15.1.0'**
**implementation 'com.mindorks:placeholderview:0.2.7' for the Navigation Drawer layout**
**implementation 'com.android.volley:volley:1.0.0'**
**implementation 'com.android.support:design:26.1.0' for DrawerLayout that creates a navigation drawer**
**implementation 'com.android.support.constraint:constraint-layout:1.1.0' for the layouts**
**androidTestImplementation 'com.android.support.test.espresso:espresso-core:2.2.2' for testing**

I used Firebase to implement the chat and 'com.android.volley:volley:1.0.0' for the for Login and Register Activities. If it clicks on Login, corresponding data will be taken from URL using Volley request StringRequest(Request.Method.GET, url, new Response.Listener(). I have first created two handles of Firebase namely reference1 and reference2. When user clicks on sendButton sendButton.setOnClickListener will get implemented and required message and user will get pushed into reference1 and reference2 which will be automatically get updated in the firebase database.

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

I have added at the top-level of build.gradle a reference to the google() repo or to maven { url "https://maven.google.com" }

I used Firebase, by connecting my app to the Realtime Database, enabling my authntification by mail for the users registration.

For enabling Google APIs or Firebase services I added to the buld.grandle: 'com.google.gms:google-services:3.3.1'

**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**

- Configure libraries
- Create the Menu
- Create the Main Activities
- Connection with Firebase

**Task 2: Implement UI for Each Activity and Fragment**

- Build UI for MainActivity
- Build UI for the Descrpitive Activity named About
- Build UI for the Projects Activity
- Build UI for the Volunteering Activity (Become a Volunteer)
- Build UI for the Donation Activity
- Build UI for the Chat (Chat, User, Login, Register)
- Build UI for the Contact Activity

**Task 3: Your Next Task**

Describe the next task. For example, “Implement Google Play Services,” or “Handle Error Cases,” or “Create Build Variant.”

- Create 2 items at the menu: rating and sharing
- Implement the code for the Rating and Sharing Activities

**Task 4: Your Next Task**

- Fixing bugs and errors
- Extracting strings
- Polish the UI

**Task 5: Your Next Task**

- Give the app for testing
- User experience-changing some textColor and toast messages

---

- The app conforms to common standards found in the Android Nanodegree General Project Guidelines
- App is written solely in the Java Programming Language
- App utilizes stable release versions of all libraries, Gradle, and Android Studio.
- App stores data locally by using Firebase Realtime Database.
- App provides a widget to provide relevant information to the user on the home screen (see *Figure Widget1)*
- App regularly pulls or sends data to/from a web service, app updates data in its cache at regular intervals using a JobDispatcher.
- If no data/internet is there it will be a toast message telling you that you are not connected to the internet. For this I used ConnectivityManager
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.