System Design

JohnCena | Sprint 2

Pivotal Tracker

» **Ugandan Knuckles**

Table of Contents

Abstract

Environment Interaction

We provide a view for interaction by a single agent—the user. We employ user-input (which may include gestures or text-input) to trigger events such as to query and scrape web-pages. This means that, depending on user input, we can identify which information must be parsed in such a manner as to limit unnecessary processing. We depend on APIs to be available, running, and dependable for our app to work consistently. We chose to use Android Studio to implement this app, which restricts the software to only run on Android-compatible hardware. Some information is stored in Firebase, Android's Database system, so we require Firebase to be available for our app to function as well.

System Architecture

Our system is divided into 3 major parts. This is our MVC (Model, View, Controller). We have visualized models that match our design, and put together interfaces which control how the view displays based on the state of the model. The system architecture diagram on page 6 is the direct product of the MVC pattern.

System Decomposition

Our listener classes act as the controllers that connect the views to the models. The listeners listen for events and adjust the views accordingly. The model is where we query all the data. When we get user input, the model handles the action and finds charity information. Some of this information includes the charity logo, purpose, social media links, and revenue information. The views are the UI components that the user will see and are the basis of all user interactions. To handle errors, we are setting up error messages with try-catch blocks to predict runtime vulnerability. These messages are meant to be purely instructional so as to explain to the user what the issue is. Aside from system errors, user errors may also arise in our app, such as invalid user input. To deal with this, we have chosen to use the safety net of the API's we are using—which deal with invalid input on our behalf.

CRC Model

MapActivity	
• Knows radius around current location • Shows Charities on the map as markers (based on radius) • Acquires charity information	**LocationSuccessListener** **CameraMoveListener** **PlaceSelectListener** **SummaryLogoScraper** **UrlAndDonationLinkScraper** **User** **MarkerClickerListener**

LocationSuccessListener	
• Gets the current location of the user	**User**

CameraMoveListener	
• Moves the map based on user gestures	**User**

SummaryLogoScraper	
• Parse Summary and Logo information for Charity	

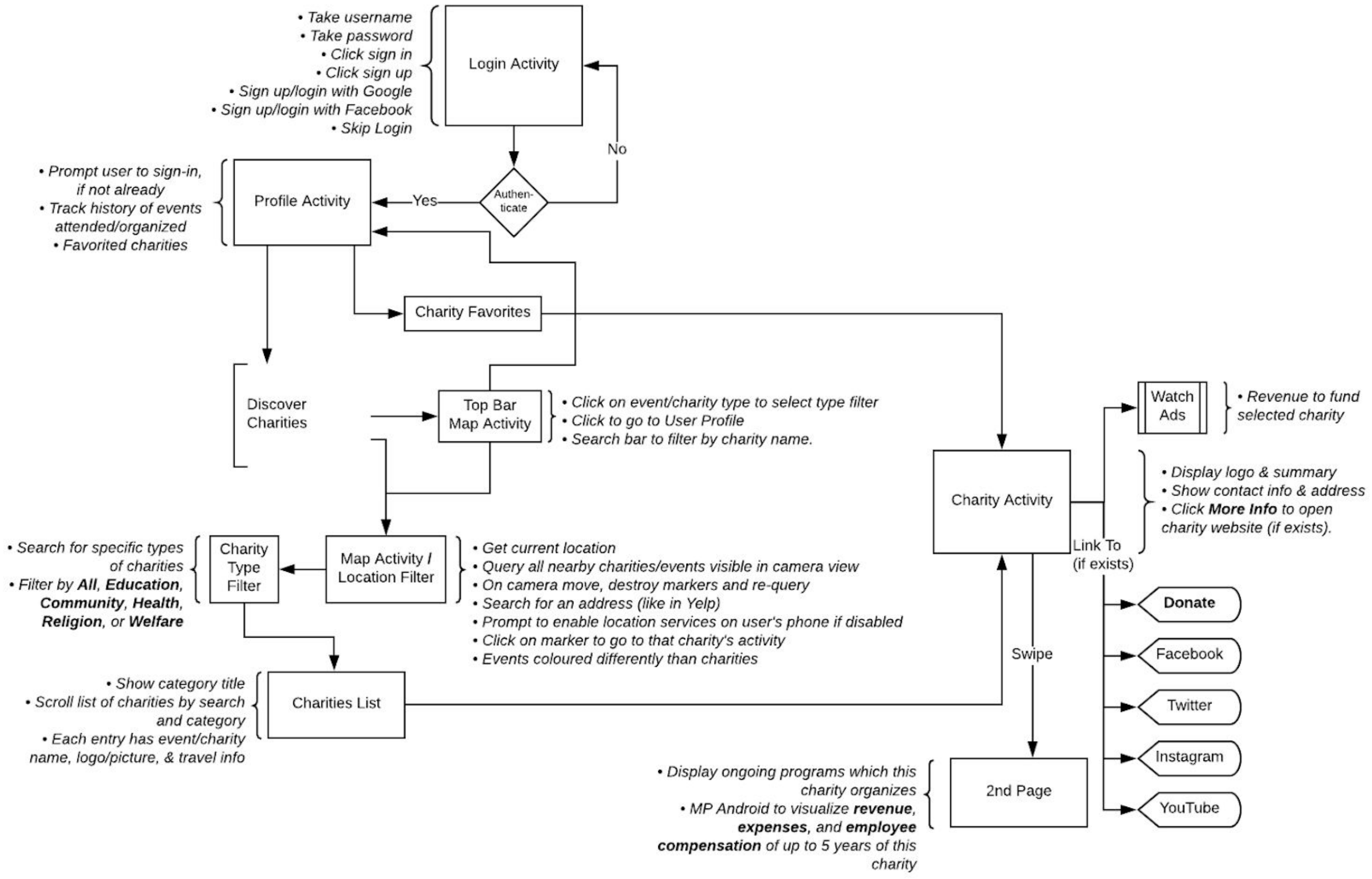
UrlAndDonationLinkScraper	
• Parse Donation and Links information for Charity	

User	
• Know personalized account information	

MarkerClickerListener	
• Find directions from current position to charities currently displayed on the map	**User**

System Architecture

The following diagram demonstrates the interconnection of components, subdivided by state of the app in relation to the user-interface.


• Take username
• Take password
• Click sign in
• Click sign up
• Sign up/login with Google
• Sign up/login with Facebook
• Skip Login
Login Activity
No
Authen-ticate
Yes
• Prompt user to sign-in, if not already
• Track history of events attended/organized
• Favorited charities
Profile Activity
Charity Favorites
Discover Charities
Top Bar Map Activity
• Click on event/charity type to select type filter
• Click to go to User Profile
• Search bar to filter by charity name.
• Search for specific types of charities
• Filter by **All**, **Education**, **Community**, **Health**, **Religion**, or **Welfare**
Charity Type Filter
Map Activity / Location Filter
• Get current location
• Query all nearby charities/events visible in camera view
• On camera move, destroy markers and re-query
• Search for an address (like in Yelp)
• Prompt to enable location services on user's phone if disabled
• Click on marker to go to that charity's activity
• Events coloured differently than charities
• Show category title
• Scroll list of charities by search and category
• Each entry has event/charity name, logo/picture, & travel info
Charities List
Charity Activity
Watch Ads
• Revenue to fund selected charity
• Display logo & summary
• Show contact info & address
• Click **More Info** to open charity website (if exists).
Link To (if exists)
Swipe
Donate
Facebook
Twitter
Instagram
YouTube
• Display ongoing programs which this charity organizes
• MP Android to visualize **revenue**, **expenses**, and **employee compensation** of up to 5 years of this charity
2nd Page