Sprint 1

JohnCena

Pivotal Tracker: Ugandan Knuckles

System Design

TABLE ON CONTENTS

Contents

TABLE ON CONTENTS....1
Summary....3
CRC CARDS....4
System Architecture....6

Summary

Environment Interaction:

The user interacts with the view we provide. From user input (this may include gestures or text input), we trigger events that query and scrape webpages. This means that the information we choose to parse is dependent on user input. We also depend on APIs to be available and running for our app to work. Because we chose to implement this with Android Studio, only Android phones would be able to run the app. Some of the information are going to be stored in Firebase, Android's Database system. We need Firebase to be available for our app to work 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. We also have interfaces that control how the view looks based on the state of the model. A diagram is provided on page 5.

System Decomposition

Our listener classes acts as the controllers that connect the views to the models. The listeners listen for events and adjusts 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 will be the basis of what the user will be interacting with. To handle errors, we are planning to set up error messages with various try and catch blocks to predict runtime vulnerability. These messages are to be meaningful so the user would know what went wrong. Aside from system errors, user errors may also arise in our app. This may include invalid user input. To deal with this however, we have chosen to use the safety net of the APIs that we chose to use. They deal with invalid input on our behalf.

CRC CARDS

MapActivity

Responsibilities:

-Knows its radius around the current location

-Shows Charities on the map as markers (based on radius)

-Gets charity information

Collaborators:

-LocationSuccessListener

-CameraMoveListener

-PlaceSelectListener

-SummaryLogoScraper

-UrlAndDonationLinkScraper

-User

-MarkerClickerListener

LocationSuccessListener

Responsibilities:

-Gets the current location of the user

Collaborators:

-User

CameraMoveListener

Responsibilities:

-Moves the map based on user gestures

Collaborators:

-User

SummaryLogoScraper

Responsibilities:

-Get the Summary and Logo information

UrlAndDonationLinkScraper

Responsibilities:

-Get the Donation and Links information

User

-Responsibilities:

-Know its account information

MarkerClickerListener

Responsibilities:

-Get direction from the current position to charities currently displayed on the map

System Architecture