

Jaago

Design Document

Team

Sharad Gupta - 201101199
Aman Jain - 201101132
Nishant Krishan - 201101139
Rishabh Raj - 201101211
Nikhar Agrawal - 201101217
Sumol Lodha - 201101003

Specification

The application aims to provide momentum to social movements/struggles by raising public awareness by providing a platform for people to contribute to social causes by organizing events. It serves as a database for people to find the causes they most identify with.

Design

This social awareness app is a hybrid application since it runs on an Android Device but uses an external database hosted on a linux server over the World Wide Web to fetch the data to display to the end-user. We contact a server URL, which upon visiting runs an SQL query over a database, and returns a JSON object to the application which is then parsed and displayed accordingly.

Our application recognizes the social movements/struggles as 'Causes'. We have a central server from which our application fetches the data. All the data is stored in a database using MySQL. We fetch the data by calling PHP pages on web browsers that returns the information as JSON objects. With each cause, any logged in user has the ability to create events. The event functionality allows users to meet and discuss about these movements/struggles.

Home-Screen

On the home screen for the end-user, there is a list of Causes displayed, for each cause, a title and a short description is displayed. We have a button to Create a New Cause as well as filter which can filter all the stored causes on the basis of the cities and display them in scroll view to the user.

A View for each “Cause”

Each of the list item is clickable and takes the end-user to a separate screen layout, which displays the number of upvotes on the “cause”, the complete description of the cause and comments made by other end-users. Any user can add a new comment or upvote or downvote the cause.

Clicking on the upvote or downvote button runs a query on the backend to update the database, which further updates the number of upvotes on the “cause”.

On this screen, there is also an events link which displays all the events related to a particular cause which took place. If a meeting was scheduled for a particular cause, it will be visible on the events link for each cause. Also, users can synchronize the events with their Google Calendar.

“Create Cause” view

We have a separate screen layout for “Create a New Cause”. There are four text boxes for taking input, namely, “Title”, “Description”, “City”, “Tags”. We can reach this screen clicking the Create Cause button on the home screen.

The list view in each screen is populated by visiting a server URL, receiving the JSON response, parsing it and rendering in a proper fashion.

Backend - MySQL

We have a MySQL database that stores all the information on a central server in the following fashion:

- Users
- Causes
- Events (Primary key of Causes is a foreign key in Events)
- Comments

Backend - PHP

We have PHP files that return the required information as JSON objects or inserts the POST data into MySQL.

- get_*.php lists the required info by using GET arguments passed and running SQL queries on them.
- create_* php files insert POST data that it receives from the android application in the database.
- some other PHP files that deal with filtering results, updating records etc.

Frontend - Login

- Our launcher activity is a simple login/signup page. It verifies the credentials from the database.
- On successful authentication, an intent is used to call the home activity and the user name is passed as extra info.
- The home activity by default displays the global top rated causes.

Frontend - Home and Cause details

- The home activity also allows us to filter causes based on the city and tags.
- On clicking a cause, a new activity is launched which displays the details of the cause.
- Along with each cause, an activity is called that displays all the comments related to the cause.

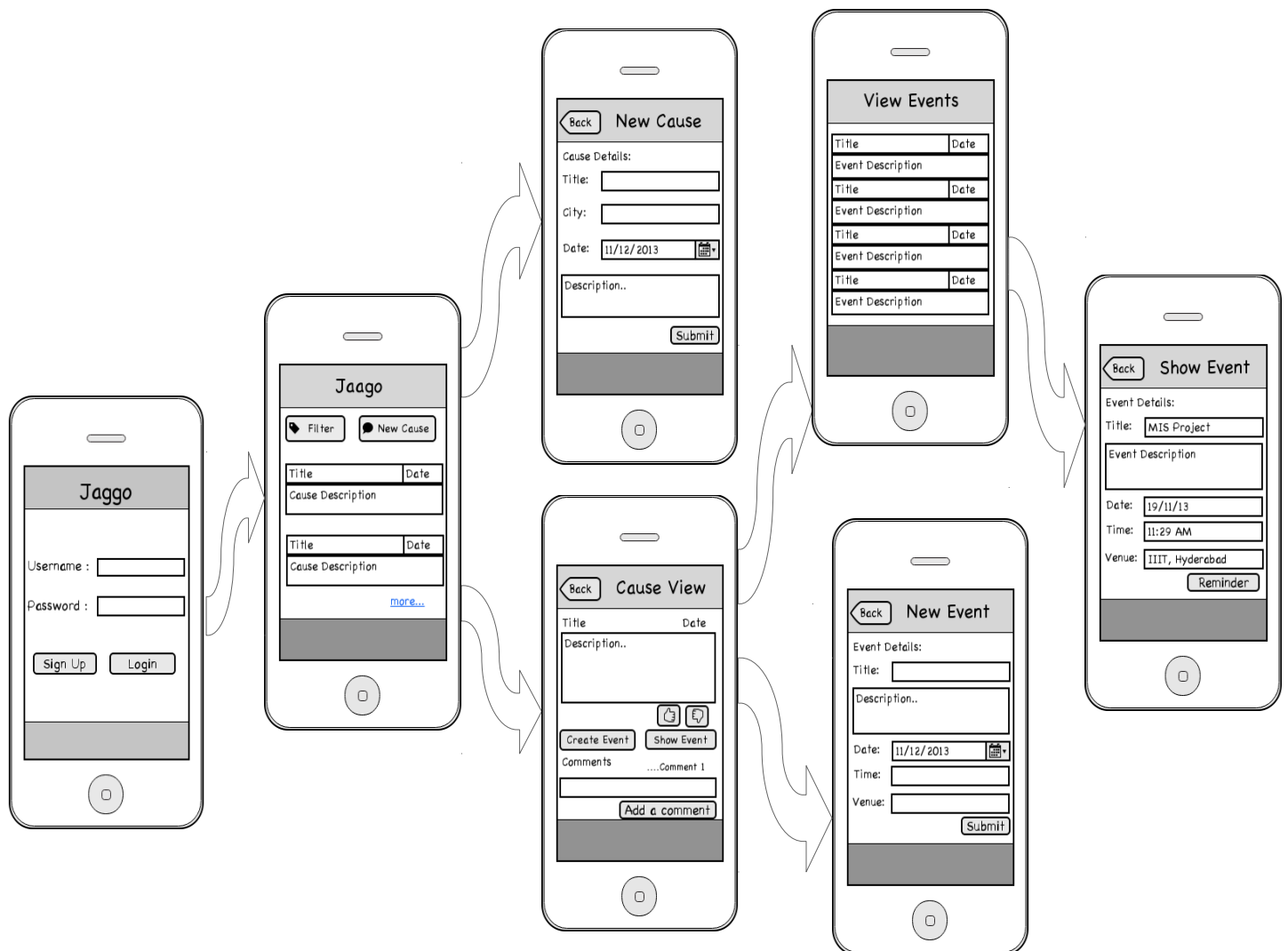
Frontend - Events

- With each cause we have associated events. Any logged in user can create

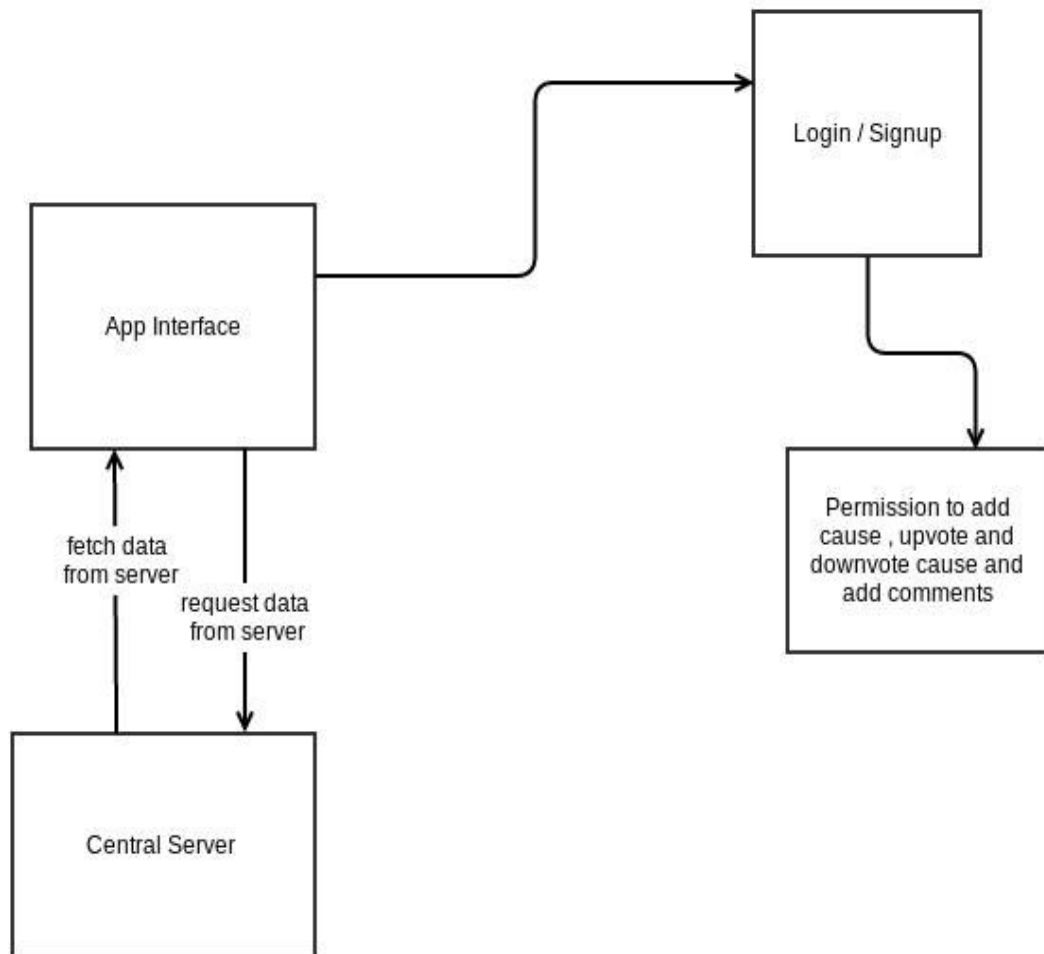
an event that would be associated with the cause.

- On the cause description page, we have buttons for creating an event and listing all events associated with the cause.
- The OnClickListener for creating an event calls a new activity which then inserts data into database via PHP using POST data.
- The button for list all events launches another activity. The cause_id is passed as extra information to identify all the events listed with the cause.
- We can then click on any event and an activity is launched displaying information about the event.

UI Flow



Architecture



OS/API Specific Details

- Android API Minimum Level - 10
- Linux Operating System running the LAMP stack

Note: LAMP stack stands for **L**inux the operating system, **A**pache HTTP server, **M**ySQL database management system, **P**HP the scripting language.