**Event App Documentation**

**Overview:**

The LightSys Events App is an app used for the organization and distribution of event information for ministry events. Users can view notifications, contacts, event schedules, prayer partners, housing, and information pages.

**Main Features:**

* The app allows the user to scan a QR code. Using the URL retrieved from the QR code, the app will load event information from a JSON file. This JSON file is created by the LightSys web app from the information that the event host provides.
* The app allows the user to have five scanned events at a time. When a sixth event is scanned, the least recently used event is removed. The user can also delete a scanned event with a long press. Only one event’s data is loaded at a time.
* If there are no scanned events, the app is in a state that asks the user to scan an event. In this state, the about page and settings activity are both accessible.
* The user can change how often they would like to receive notifications. This ranges from never to every 60 minutes. The user can also select ‘auto’ and the ‘event default’. The user can also manually refresh the event.
* The user can change their time zone settings. The user can select the ‘On-Site’ time zone, their current location, or select a custom time zone from a sorted list.
* The user can view several views, depending on what the event host has provided in the web app. The current views available are: Notifications, Contacts, Schedule, Housing, Prayer Partners, and Information Pages. An ‘About’ page is also present to provide information about the app.
* The titles of these views can be changed if specified by the host in the web app.
* The schedule can adjust based on time zone changes. The schedule will also fill the current time frame yellow. The schedule can have different color highlights for different types of activities if specified by the host in the web app.
* Users can tap on a contact’s information and can choose to call the associated number by opening their phone app or get directions to the associated address by opening their directions app. This can be done from either the contacts view or the housing view.
* An event logo can be displayed if the event host provides a logo in the web app.
* Links are clickable.

**Future TODOs:**

* Implement horizontal fling in the schedule
* Allow specification of calendar scale and time compression factor.
* A side bar with descriptions and links for resource items.
* In-app maps for event locations. This would include a list of maps, showing the device’s current location, zooming/panning, map icons for places like restrooms, marked location of current schedule items. This would also include the ability to view unmarked maps when location is disabled. This could only be implemented if the web app was updated to include this functionality.
* In-app airport transportation information. This would show drivers, passengers, times, and vehicle descriptions. This could only be implemented if the web app was updated to include this functionality.

**Issues/Bugs:**

* Some buttons may react slowly and may require multiple taps on some devices.
* The app will crash when initially opened on Android Marshmallow devices.

**App Structure:**

An event host can use the web app to create an event. The information the host provides in the web app is put into a JSON file. This JSON file can be accessed with a URL. The web app also generates a QR code leading to the URL. This QR code can be scanned by the mobile app. When the QR code is scanned, the app reads the JSON file. Much of the JSON’s event information is saved in a local database. In the local database, the information is stored in tables such as a general info table, contacts table, color table, schedule table, housing table, information pages table, and navigation titles table. These tables are cleared before a different event is loaded. Other saved information tables such as the scanned events table are not cleared when a different event is scanned because this information is used across all events. The saved information is accessed by their respective views to be formatted and displayed the app’s user.

The app has five activities. These activities include two primary activities: MainActivity.java and SettingsActivity.java. Two other activities are SettingsRecycleView.java and TimeZoneSelectionView.java which both deal with the selecting of time zones. The launchQRScanner.java activity is used for scanning QR codes.

The Main Activity is used to access several fragment views. These views include the WelcomeView, ContactsView, ScheduleView, HousingView, PrayerPartnerView, InformationPageViews, and the AboutPageView. From the main activity and its fragments, the user can access the refresh button and the scanned events dropdown. The Main Activity also allows the user to navigate to the Settings Activity, as well as the QR scanner.

The Settings Activity allows the user to change the notifications refresh rate, as well as the user’s time zone settings. The user can also access recyclerviews for the selection of custom time zones. The user can also navigate back to the main activity from the settings activity.