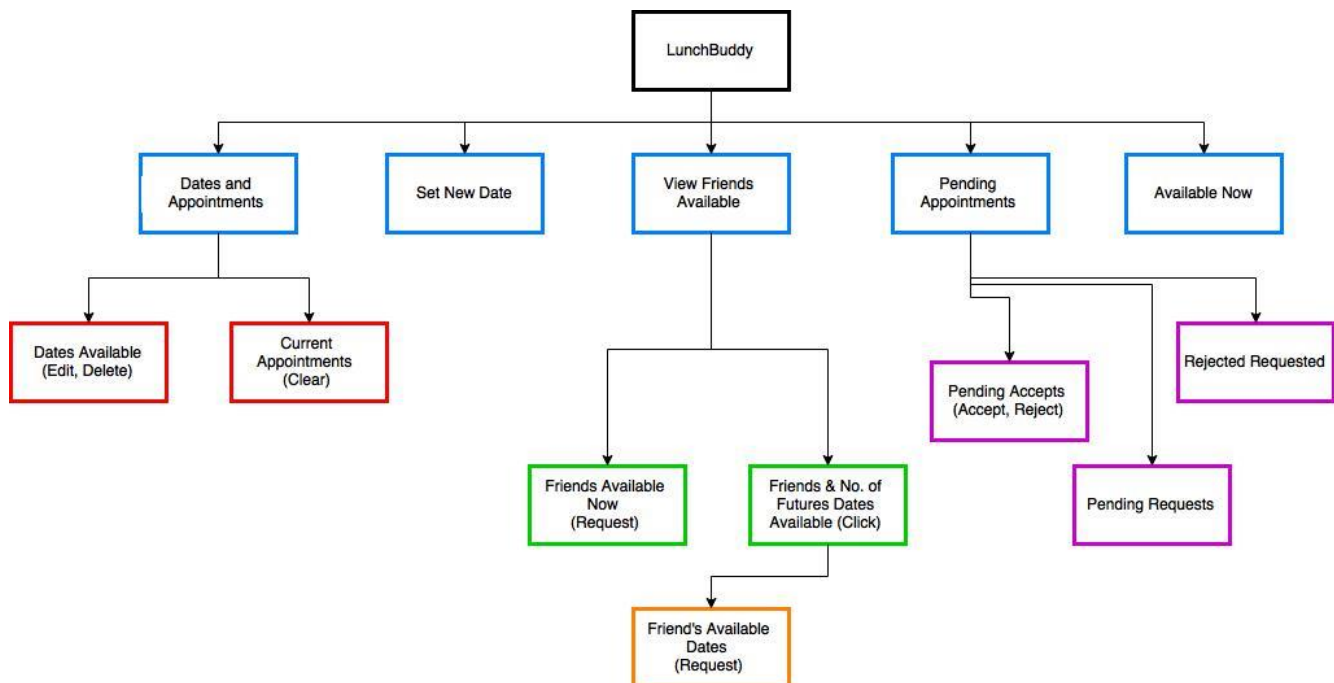


Summary

Lunch Buddy is a networking Android application programmed in Android Studio that aims to help students and office workers make plans for lunch within their social networks. The app will allow users to signal their availability to meet either immediately or at a future date, as well as to send requests to their Facebook friends who signal that they are available.

User Interaction



The above diagram shows the various components of the app, with each color representing a different screen available to the user.

Screen	Description
Home	<p>The home screen has 4 different buttons, which allows the user to navigate to the respective portions of the app which he wishes to access.</p> <p>The <i>Available Now</i> button allows the user to toggle his current status between “available” and “unavailable”.</p>

Dates and Appointments	<p><i>Dates Available</i>: displays all the future dates the user has set available</p> <ul style="list-style-type: none"> - Edit: allows the user to edit the particular entry. Clicking this directs the user to the <i>Set New Date</i> activity, where he is allowed to key in a new date. - Delete: allows the user to delete the particular entry. <p><i>Current Appointments</i>:</p> <ul style="list-style-type: none"> - Clear: allows the user to clear the particular entry.
Set New Date	<p>Allows the user to set and update a new date and time in which he is available. Creates a new entry when accessed from the home screen, and edits a current entry when accessed from the <i>Dates Available</i> ListView.</p>
View Friends Available	<p><i>Friends Available Now</i>: displays friends that have their “available now” button turned on, signaling that they are available to meet in the near future.</p> <ul style="list-style-type: none"> - Request: sends a request to that particular friend. <p><i>Friends & No. of Future Dates</i>: displays all the user’s friends that have the app installed, as well as the number of entries in that friend’s “Dates Available” table.</p> <ul style="list-style-type: none"> - Clicking on an entry in this ListView opens up a new page, <i>Friend’s Available Dates</i>
Friend’s Dates Available	<p><i>Friend’s Available Dates</i>: shows the specific dates that friend has set available.</p> <ul style="list-style-type: none"> - Request: Sends a request to that friend and date
Pending Appointments	<p><i>Pending Accepts</i>: shows the current requests made to the user by his friends. If requested from the <i>Friends Available Now</i> page, shows “immediate” next to his friend’s name instead of a specific date.</p> <ul style="list-style-type: none"> - Accept: accepts that particular request and moves it to the

	<p>user's <i>Current Appointments</i></p> <ul style="list-style-type: none">- Reject: rejects that particular request and moves it to his friend's <i>Rejected Requests</i> ListView. <p><i>Pending Requests</i>: shows the requests that the user has made but have neither been accepted nor rejected by his friends. If requested from the <i>Friends Available Now</i> page, shows "immediate" next to his friend's name instead of a specific date.</p> <p><i>Rejected Requests</i>: shows the requests that the user has made, but have been rejected by his friends. If requested from the <i>Friends Available Now</i> page, shows "immediate" next to his friend's name instead of a specific date.</p> <ul style="list-style-type: none">- Clear: allows the user to delete the particular request from his history.
--	--

Technical Documentation

Activities and Fragments

- CalendarActivity
- FriendDetailActivity
 - o FriendDetailFragment
- FriendDisplayActivity
 - o FriendDisplayFragment
- MainActivity
- PendingActivity
- ProfileActivity

Database and Content Provider

- DatesAvailDatabase
- DatesAvailListProvider

Custom Adapters

- ConfirmedListAdapter
- DatesAvailCursorAdapter
- DetailListAdapter
- FacebookListAdapter
- FriendsNowListAdapter
- PendingListAdapter
- RejectedListAdapter

Helper Classes

- Application
- FacebookFriend

Launching the App

Opening LunchBuddy runs Application and MainActivity, which initializes the basic set of permissions from Parse and Facebook for the app to run. MainActivity automatically searches for an existing Facebook app login and uses that ID, else it directs the user to a Facebook login page in order to access the rest of the app. The saveFacebookId() method retrieves the user's Facebook ID and stores it as a value in the ParseUser object associated with the user, which will the app uses to establish future relations without re-querying Facebook.

The Available Now button is a toggle button that changes a Boolean value associated with the current Parse User on the Parse database. Clicking "Available" changes the value to true, signaling that the user is available to meet and showing up on his friends' "Friends Available Now" list. Clicking "Unavailable" changes this value to false, but does not override appointment requests that have already been made.

Dates and Appointments

Clicking "Dates and Appointments" launches ProfileActivity, which contains two ListViews: current appointments and dates available.

Current Appointments queries the “AcceptedAppts” class in the Parse database, and displays it if the “Owner” field in “AcceptedAppts” corresponds to the user’s Parse User ID. It uses ConfirmedListAdapter to display the objects retrieved, and includes a “clear” button, which deletes the object from the database when clicked.

Dates Available implements a local SQLite database using DatesAvailDatabase. The app first queries Parse and downloads objects created by the user in Parse’s “Dates Available” class into the local database. Using the content provider customized in DatesAvailListProvider, the app then queries the SQLite database dates.db and adapts it to ListView for the user via DatesAvailCursorAdapter. Clicking edit creates an intent and launches CalendarActivity, while clicking delete launches a parse query that deletes the object from the online database. Clicking refresh simply relaunches the activity and refreshes the ListView with the latest data from Parse.

Set New Date

Set New Date opens up CalendarActivity, which checks if which parent activity it was opened from. If opened from MainActivity, the user creates a new Parse Object and uploads it to the “DatesAvailable” class in Parse. Opening CalendarActivity from the “Dates Available” ListView allows user the user to modify the existing object, which has been passed between the activity via an Intent.

Friends Available

FriendsAvailable consists of two ListViews: “Friends Available Now” and “Friends and Future Dates Available”.

Upon creation, FriendsDisplayActivity checks if the device is a large-screened device (e.g. tablet). If it is a large screen, it sets the two-pane Boolean to be “true”, which allows FriendDetailFragment to be displayed next to FriendsDisplayFragment on the tablet. Android automatically detects the screen size, and switches to use the xml file in the “layout-large” resources folder if it is a large-screened device.

FriendsDisplayFragment uses an AsyncTask to launch a GraphRequest to Facebook, in order to retrieve the list of friends who also have the app installed. For each friend, a new FacebookFriend object is created and added into the facebookIds array. Next, Parse iterates

through each friend in facebookIds by associating the user's Facebook ID with his Parse Username, and updates the dates available field in that object. Additionally, it checks the Boolean value of "Available" for each Parse User. A Boolean value of true signals that the user is available now.

The activity then updates the Friends Available Now ListView (implemented using FriendsNowListAdapter), as the Friends and Future Dates Available. To get the number of dates available, the app checks the size of the "Dates" array for each FacebookFriend object.

Friend's Available Dates

Clicking on any friend in the ListView sends an intent or bundle to FriendDetailFragment, which displays the friend's specific dates available via a DetailListAdapter. If opened in a tablet, this screen takes up the second half of the two-pane display. Else, a new screen is launched on the smartphone.

Clicking "request" sends a query to parse, instructing it to create a new object in the "PendingAppts" class that stores both the poster and requestor's name and ID.

Pending Appointments

This launches PendingActivity, which consists of 3 ListViews: PendingAccepts, PendingRequests and RejectedRequests.

Pending Accepts displays the requests that have been made by the user's friend to the user, and is implemented through PendingListAdapter. Clicking "Accept" deletes the current pending appointment (this request will no longer show under the user's friend's "Pending Requests" ListView), and creates two objects in Parse's "AcceptedAppts" class: one with the "Owner" field associated with the poster, and one associated with the requestor. Clicking "Reject" also deletes it from "PendingAppts", and moves it to the "RejectsAppts" class on Parse.

Pending Requests queries the Parse database's "PendingAppts" class, and displays all objects whose Requestor ID matches that of the current Parse User.

Rejected Requests queries the Parse database's "RejectedAppts" class, and displays all objects whose Requestor ID matches that of the current Parse User. Users have the option to clear items

in this ListView, which will delete the corresponding object from Parse well. This ListView is implemented through RejectedListAdapter.

Layout

Rotating the screen does not disrupt the current status of the app. Special views for the MainActivity and CalendarActivity were made in the resources folder “layout-land” to better optimize for landscape views. The ListView pages have linear layouts that fill the parent, and thus do not need special layouts created. In order to preserve the state of the app and prevent the app from destroying and re-starting, the line

```
android:configChanges="keyboardHidden|orientation|screenSize"
```

was added to the Android Manifest in the relevant activities.

A tablet view is catered to in the FriendsDisplayActivity and FriendsDetailActivity, allow the user to make full use of the screen.

Error Handling

The current iteration of the app requires a stable internet connection, as it constantly queries the Parse Database directly. Most activities check the present network state before launching. If no internet connected is detected, the toast error appears informing the user of this connectivity issue.

Because ParseQueries run on a separate thread, the query may not finish running before list adapters are updated in some of the activities. For example, deleting an object does not immediately update the ListView. However, if an objected signaled for deletion is queried again and Parse is unable to find it, a toast message instructing the user to refresh the page is displayed. The page can be refreshed by going back to the home page and re-opening the page again.