



Google  
**Developers**



# Building Compelling mCommerce Experiences on Android

Tips and Tricks for Higher Conversion on Android

John Stuppy / Jon Boekenoogen - Google Wallet Instant Buy Engineers  
Haisoo Shin - The Fancy: Director of Engineering



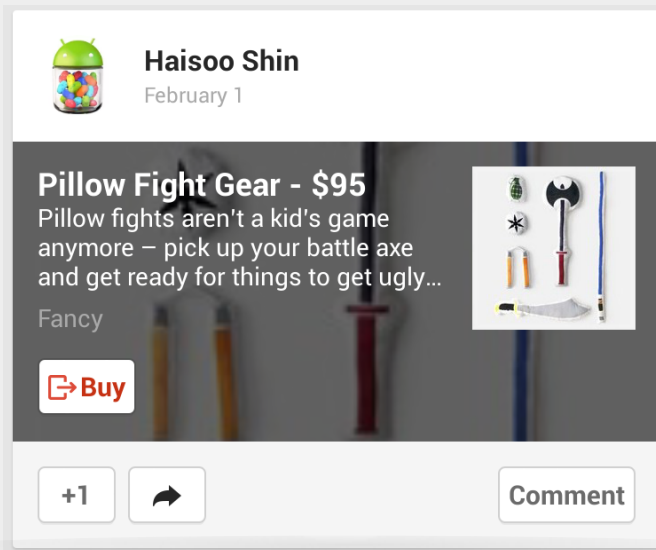
900M  
Android Devices



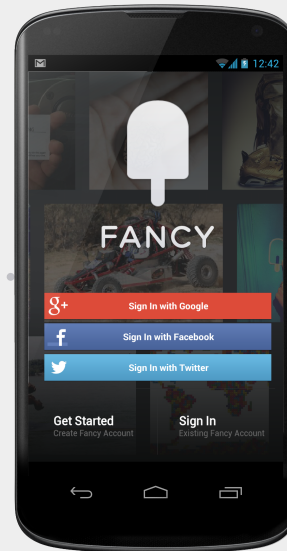
# Challenges on Mobile Commerce

97%

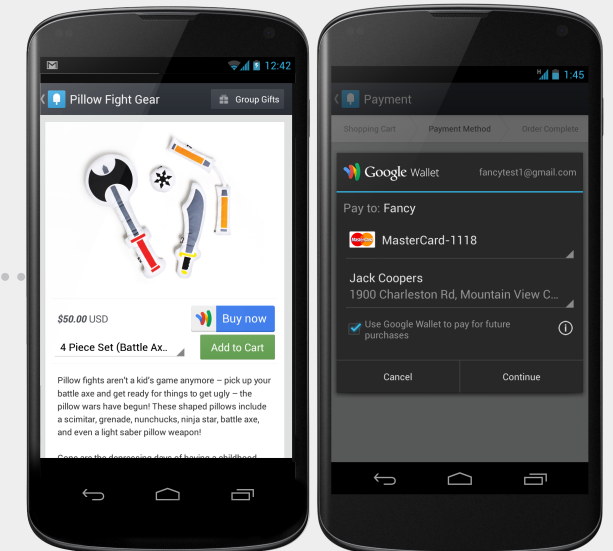
mobile cart abandonment rate<sup>1</sup>



App/Product Discovery  
Google+ Interactive Posts



Account Management  
Google+ Sign in

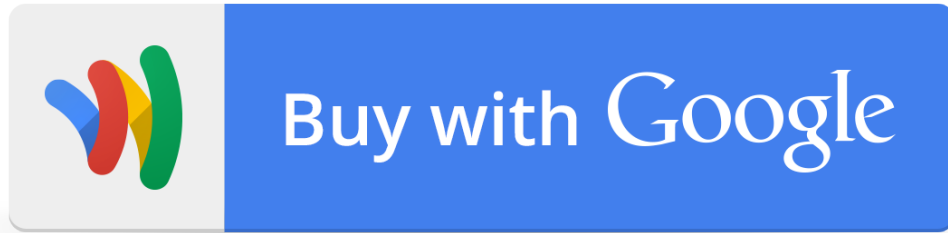


Payment  
Google Wallet Instant Buy  
Android API



<sup>1</sup> Should the 97% Shopping Cart Abandonment Rate on Mobile Devices Concern You? (SeeWhy, Oct 10, 2012).  
<http://seewhy.com/blog/2012/10/10/97-shopping-cart-abandonment-rate-mobile-devices-concern-you>

# Google Wallet Instant Buy



Faster checkout for physical goods

Skip registration and sign-in

Lightweight integration

More Secure

Free



# Avoid Cart/App Abandonment

Mobile friendly forms

Reducing latency

Security and privacy





# Mobile Friendly Forms

# Auto-Advance

JAVA

```
interface FieldValidator {
    boolean isComplete();
    boolean validate();
}

FieldValidator validator = new FieldValidator() {
    @Override
    public boolean isComplete() {
        // require at least 15 characters
        return editText.getText().length() >= 15;
    }
    @Override
    public boolean validate() {
        if (LuhnAlgorithm.passes(editText.getText())) {
            editText.setError(null /* clear error */);
            return true;
        } else {
            editText.setError("Invalid credit card");
            return false;
        }
    }
};
```






# Auto-Advance #2

JAVA

```
private static void initializeAutoAdvance(final EditText editText, final FieldValidator validator) {  
  
    editText.addTextChangedListener(new TextWatcher() {  
  
        @Override  
        public void afterTextChanged(Editable s) {  
            // check if the field is complete  
            if (validator.isComplete()) {  
  
                // if the field is valid  
                if (validator.validate()) {  
  
                    // advance to the next focusable field  
                    View nextView = editText.focusSearch(View.FOCUS_DOWN);  
                    if (nextView != null) {  
                        nextView.requestFocus();  
                    }  
                }  
            }  
        }  
    });  
}
```



# Auto Complete with Wallet / Places API

 United States (US)

Cardholder Name

Street Address

Apt./Suite

City

State

ZIP Code

1-650-555-1212



+



=

747 H

747 Howard Street, San Francisco, CA, U

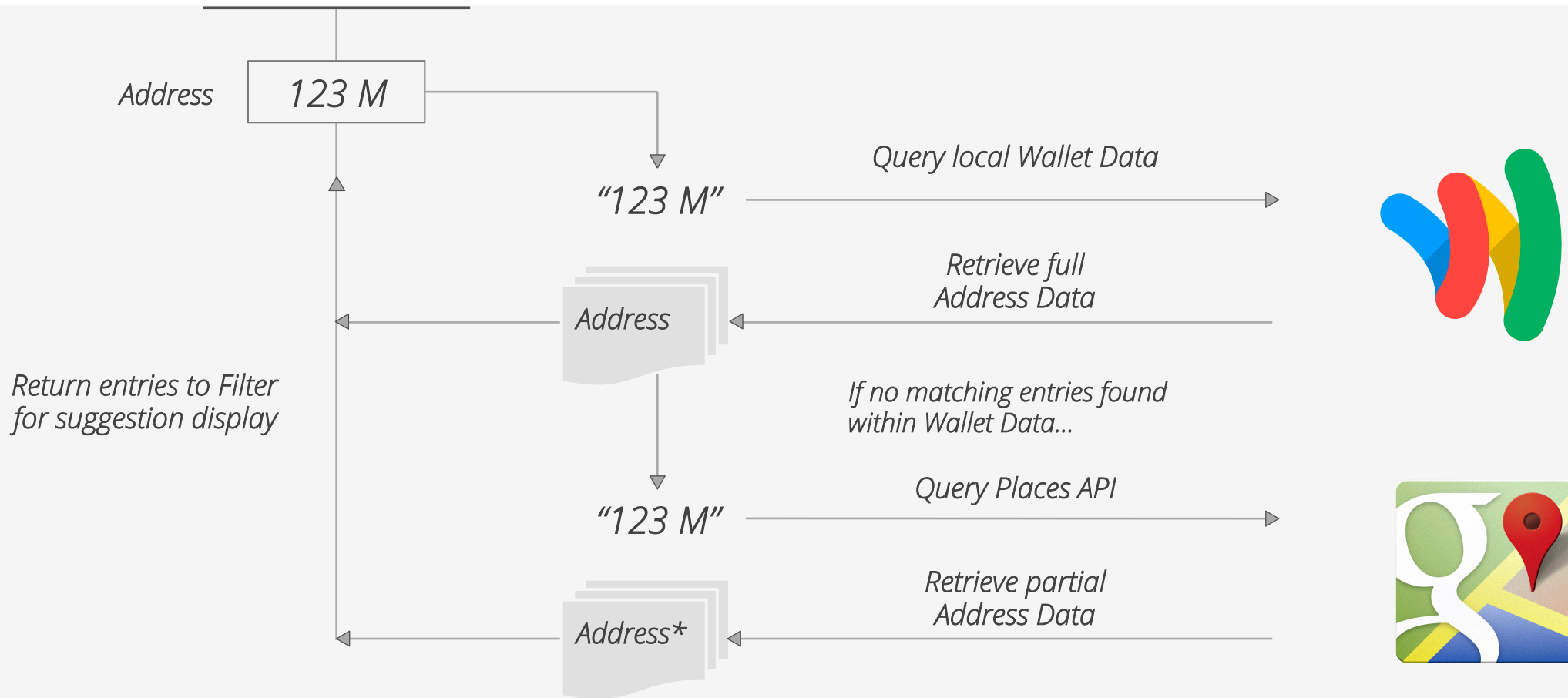
747 Harrison Street, San Francisco, CA,

747 Haight Street, San Francisco, CA, U

ZIP Code



# Auto Complete Optimizations



# AutoCompleteTextView

JAVA

```
class AddressAutoCompleteAdapter extends BaseAdapter implements Filterable {
    ArrayList<Address> mAddresses;
    @Override public Filter getFilter() { return new AddressFilter(); }
    ...
    class AddressFilter extends Filter {
        /**
         * Perform filtering for values and report the results. This method runs in a background
         * thread, so http/filesystem I/O will not block the UI thread
         */
        @Override
        protected FilterResults performFiltering(CharSequence constraint) {
            // make request to Address source(s) incl. Google Places API, local data, etc
        }
        /**
         * Alerts the UI that there are new suggestions to show.
         */
        @Override
        protected void publishResults(CharSequence constraint, FilterResults results) {
            mAddresses = (ArrayList<Address>) results.values;
            if (!mAddresses.isEmpty())
                notifyDataSetChanged();
            else
                notifyDataSetInvalidated();
        }
    }
}
```





# Reducing Latency

# On Device Caching

125ms

Inflate UI quickly

Cache data that changes infrequently



# Issues with On Device Caching



File permissions



Cache inconsistencies



Battery / network impact



# Disk Storage – Shared Preferences

JAVA

```
public class AddressStore {
    private final SharedPreferences sharedPrefs;

    public AddressStore(Context context) {
        sharedPrefs = context.getApplicationContext()
            .getSharedPreferences("addressStore", Context.MODE_PRIVATE);
    }

    public ArrayList<Address> getAddresses() {
        Map<String, ?> values = sharedPrefs.getAll();
        ArrayList<Address> addresses = new ArrayList<Address>(values.size());
        for (String key : values.keySet()) {
            addresses.add(Address.fromString((String) values.get(key)));
        }
        return addresses;
    }

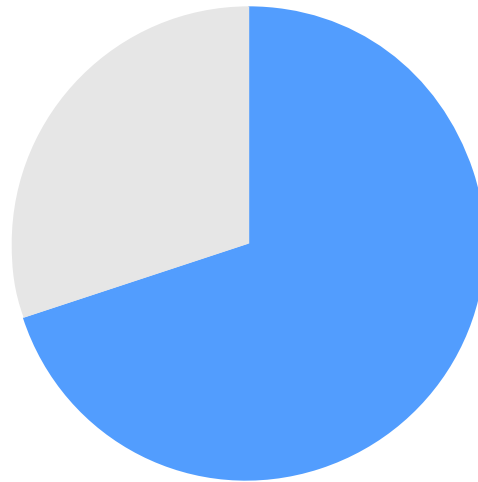
    public void writeAddresses(ArrayList<Address> addresses) {
        SharedPreferences.Editor editor = sharedPrefs.edit();
        for (Address address : addresses) {
            editor.putString(address.addressId, address.toString());
        }
        SharedPreferencesCompat.apply(editor);
    }
}
```





# Prefetch / Precompute Data

-30%



Expensive operation before its required  
Good for sequential operations



# Issues with Prefetching / Precomputing Data



Not for requests that modify state



Handle pending requests



# Prefetch Data - Auth Token

JAVA

```
public static void prefetchAuthToken(final Context context, final String emailAddress) {
    new AsyncTask<Void, Void, Void>() {
        public Void doInBackground(Void... args) {
            // GoogleAuthUtil maintains its own cache and subsequent calls read from disk.
            GoogleAuthUtil.getToken(context, emailAddress, AUTH_TOKEN_TYPE);
            return null;
        }
    }.execute();
}
```

## In some Activity:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    prefetchAuthToken(mApplicationContext, emailAddress);
}

public void onClick(View v) {
    new AsyncTask<Request, Void, Result>() {
        public Result doInBackground(Request... args) {
            // Perform network round with auth token
            String authToken = GoogleAuthUtil.getToken(mApplicationContext, emailAddress,
                AUTH_TOKEN_TYPE);
            return executeHttpPost(authToken, args[0]);
        }
    }.execute(httpRequest);
}
```





# Security and Privacy

# HTTPS

Protect sensitive  
information

Prevents man in the  
middle attacks



# Remove Logging - Proguard

TEXT

*project.properties:*

```
proguard.config=${sdk.dir}/tools/proguard/proguard-android-optimize.txt:proguard-project.txt
```

*proguard-project.txt:*

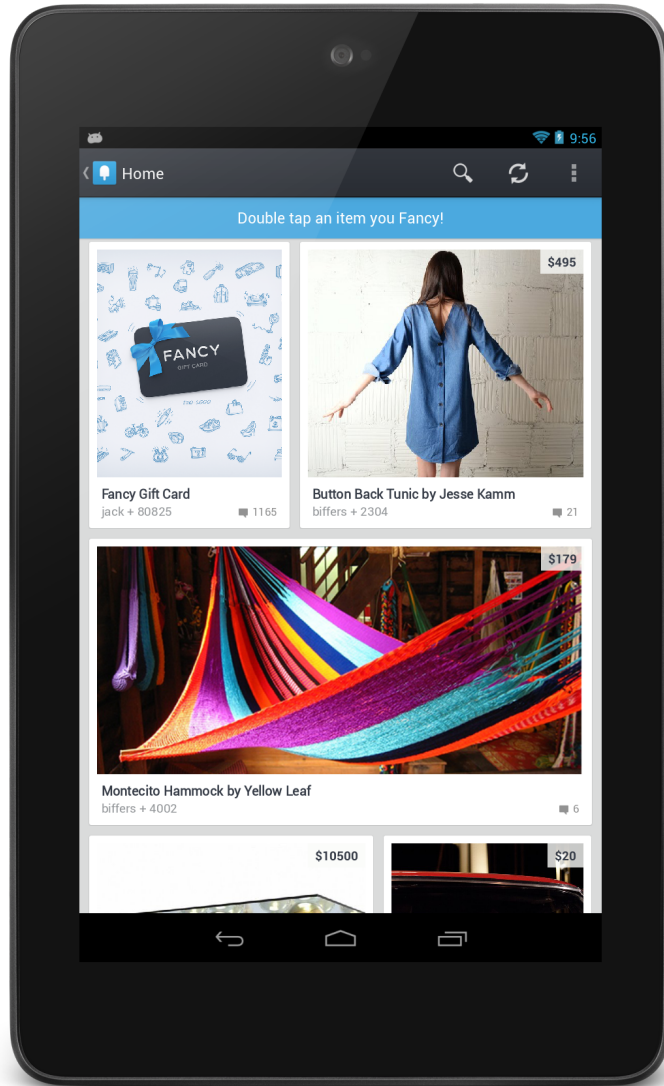
```
-assumenosideeffects class android.util.Log {  
    public static *** i(...);  
    public static *** d(...);  
    public static *** v(...);  
}
```





# Bringing It All Together Demo With The Fancy

Haisoo Shin



# Fancy

Part Store, Magazine and Wish List

Crowd-sourced collection of unique things

Discover and buy

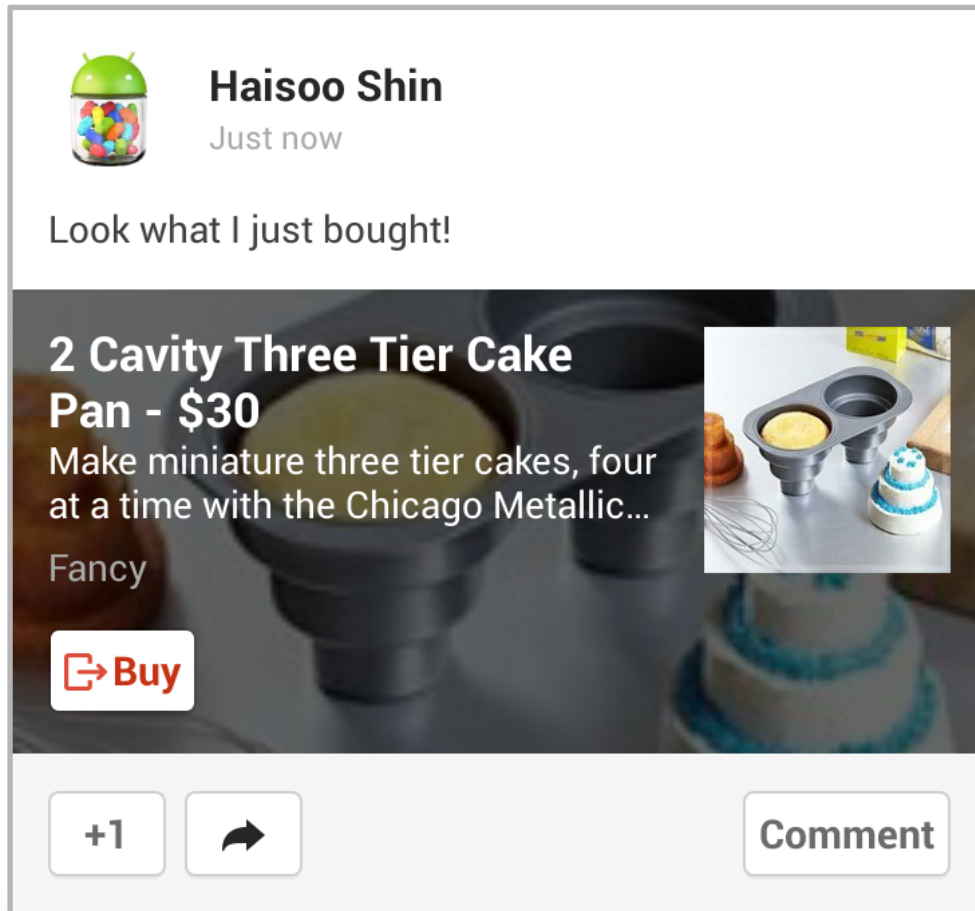
Frictionless experience from signup to checkout





# Instant Buy

with Google+ Interactive Posts



The screenshot shows a Google+ post from a user named Haisoo Shin, who has a profile picture of a green Android robot filled with colorful candies. The post text reads "Look what I just bought!". The main content is a product listing for a "2 Cavity Three Tier Cake Pan - \$30". The description says "Make miniature three tier cakes, four at a time with the Chicago Metallic..." and is marked as "Fancy". A prominent "Buy" button with a red arrow icon is visible. The post also shows interaction buttons: "+1", a share icon, and a "Comment" button. An inset image shows the cake pan being used to bake small cakes, with one finished three-tiered cake shown next to it.

Call-to-action "Buy" button

Personal promotion opportunities



# Tips on International Sales

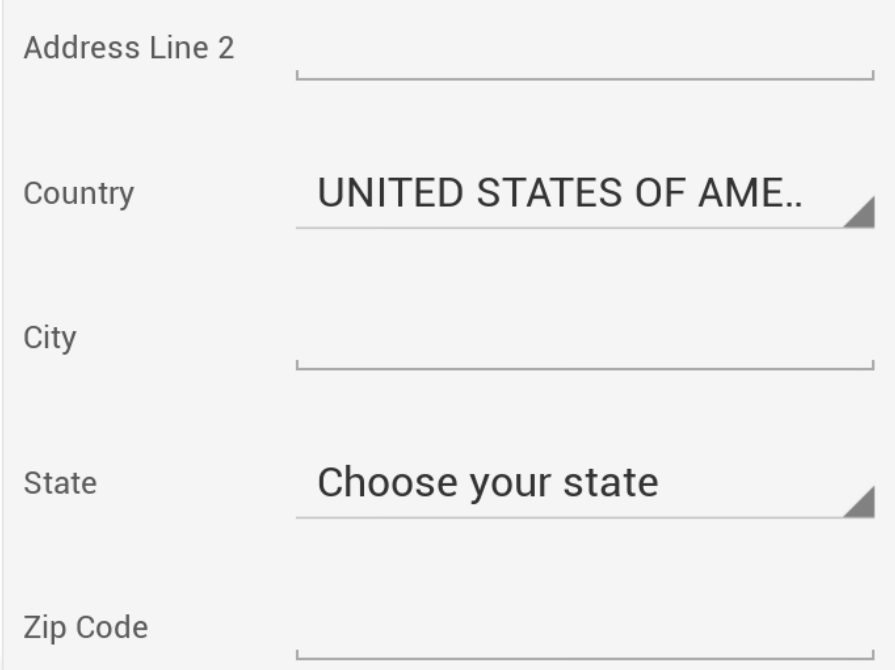
Ask for the Country before State/City

Postal Code can contain letters!

Localize your app

Enter addresses in native language

Predict user's country



Address Line 2

Country **UNITED STATES OF AME..** ▼

City

State **Choose your state** ▼

Zip Code



# Predicting User's Country

JAVA

```
public String getCountryCodeFromCarrier() {  
    TelephonyManager tel = (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);  
    return tel.getNetworkCountryIso();  
}
```

```
public String getCountryCode() {  
    // Start with carrier since it is more accurate  
    String carrierCountry = getCountryCodeFromCarrier();  
    if (!TextUtils.isEmpty(carrierCountry)) {  
        return carrierCountry;  
    }  
    // Fall back to language setting  
    return Locale.getDefault().getCountry();  
}
```



# Presume and Reduce Latency

Same payment & shipping address

Show Tax & Shipping cost early

One tap to confirm and complete

Users can change them later





Wrap up

# Wrap Up

Google Wallet Instant Buy

Real time formatting

Instant validation

Auto advancing fields

Input suggestions



# Wrap Up #2

- On device caching

- Computing / fetching data before user action

- Safe handling of user data

- Google+ sign in

- International tips



# Next Steps

Sign up for Instant Buy! - <https://getinstantbuy.withgoogle.com/>

Instant Buy Code Lab on Friday at 9 AM

Wallet office hours and developer sandbox on 2<sup>nd</sup> floor

Fancy in Google+ developer sandbox on 2<sup>nd</sup> floor

Fancy w/ Instant Buy available on the Play Store





# Thank You!

John Stuppy - [jstuppy@google.com](mailto:jstuppy@google.com)

Jon Boekenoogen - [jboekeno@google.com](mailto:jboekeno@google.com)

Haisoo Shin - [haisoo@dev.thingd.com](mailto:haisoo@dev.thingd.com)

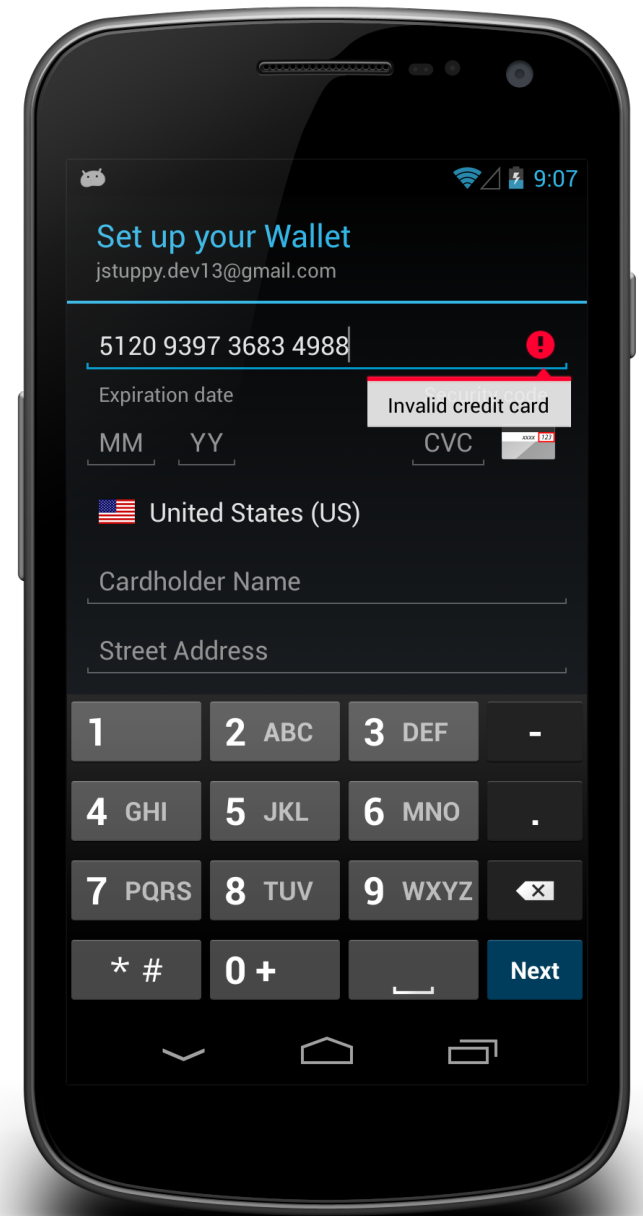
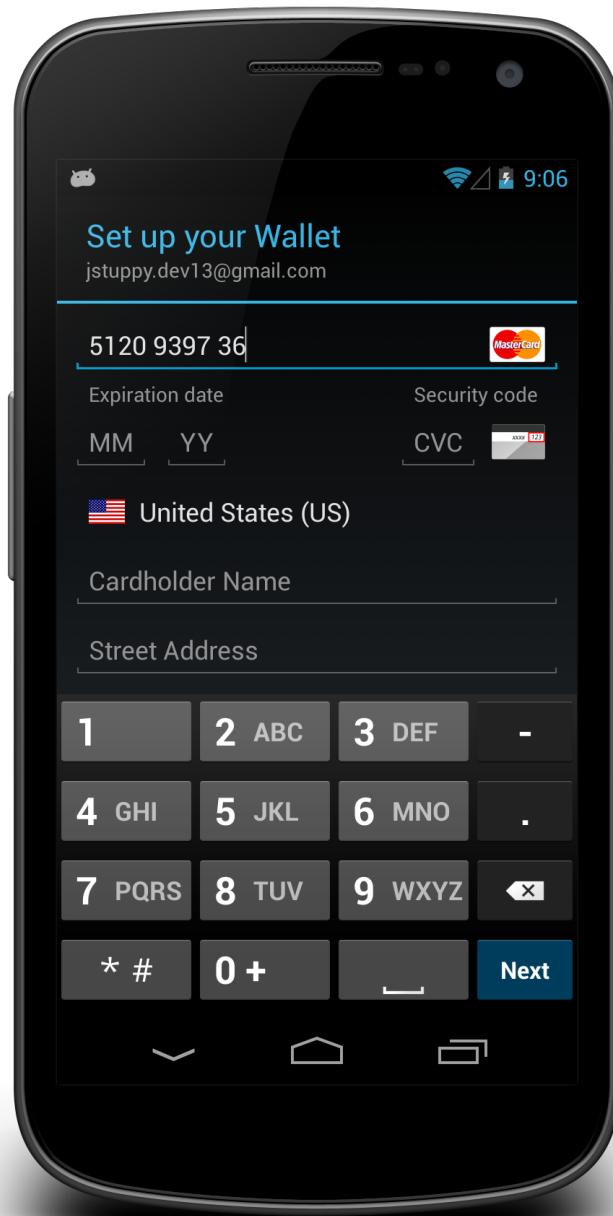
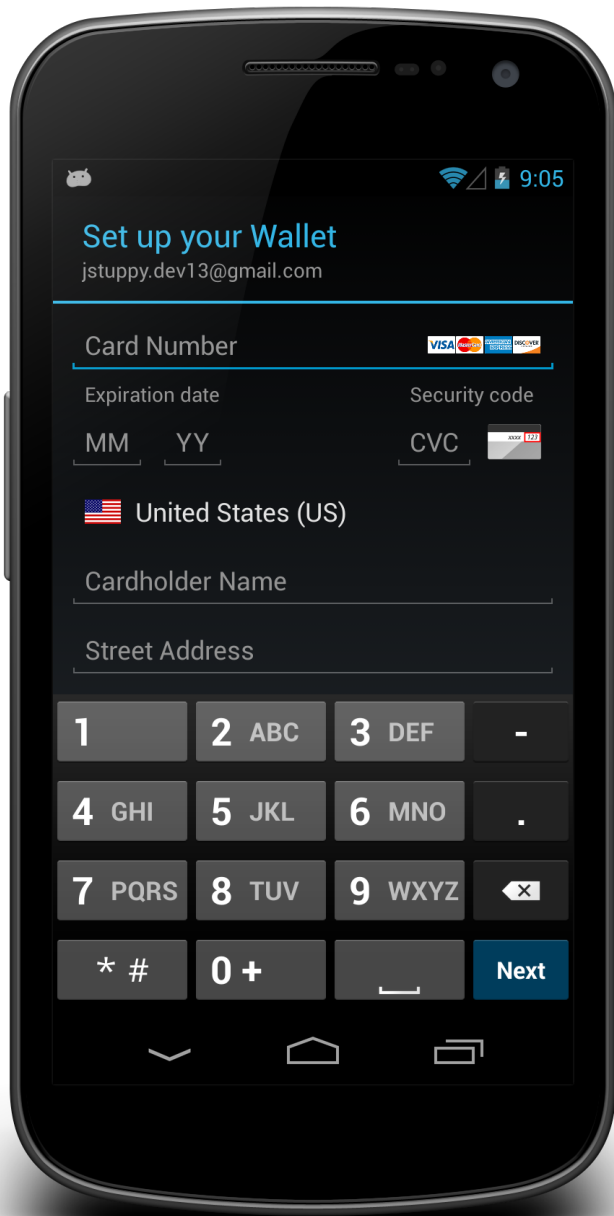


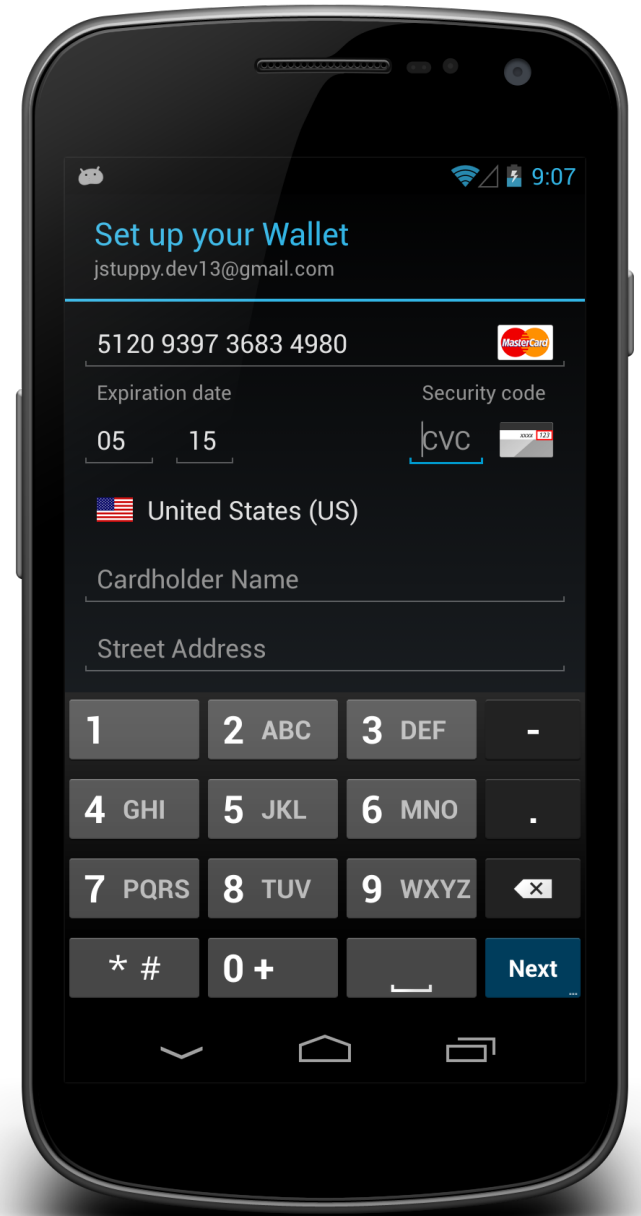
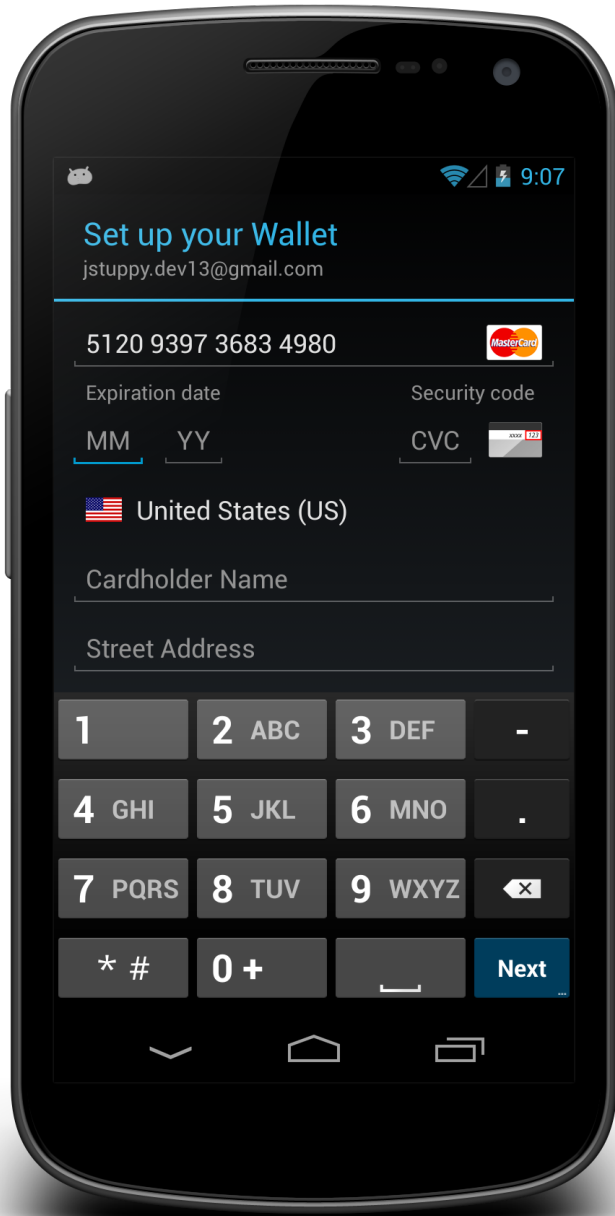
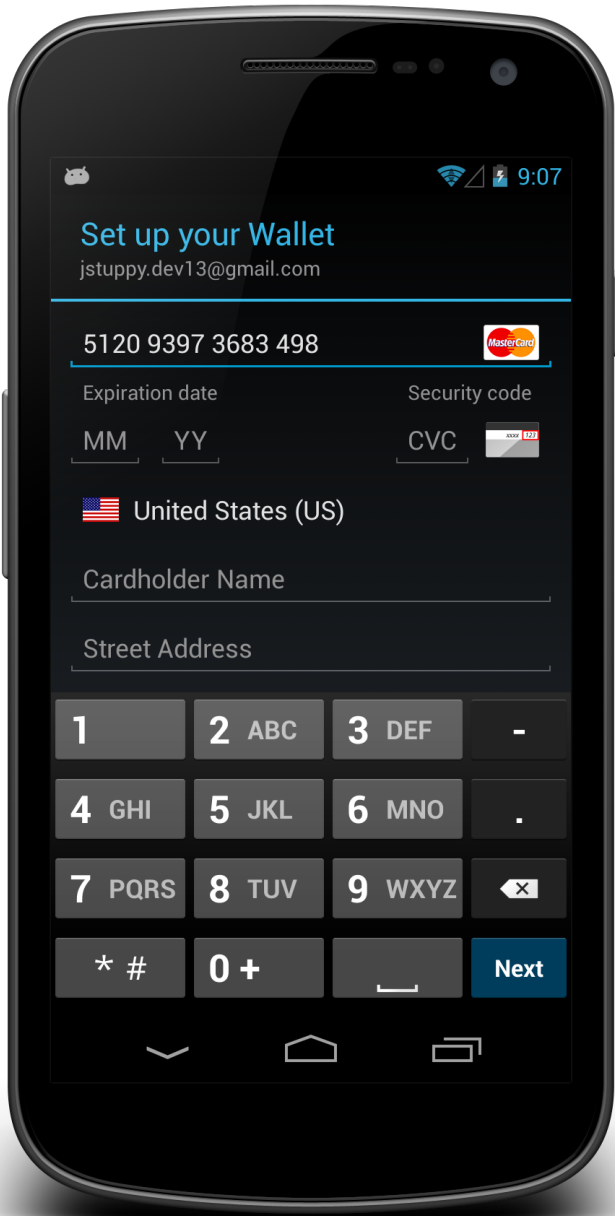


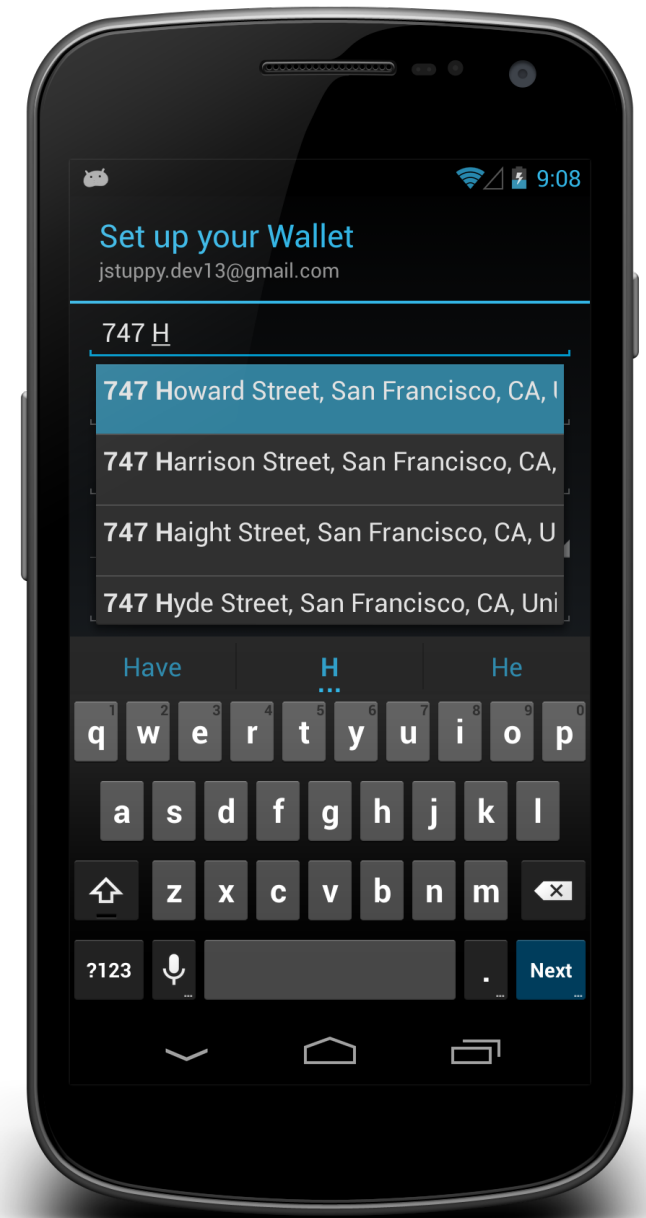
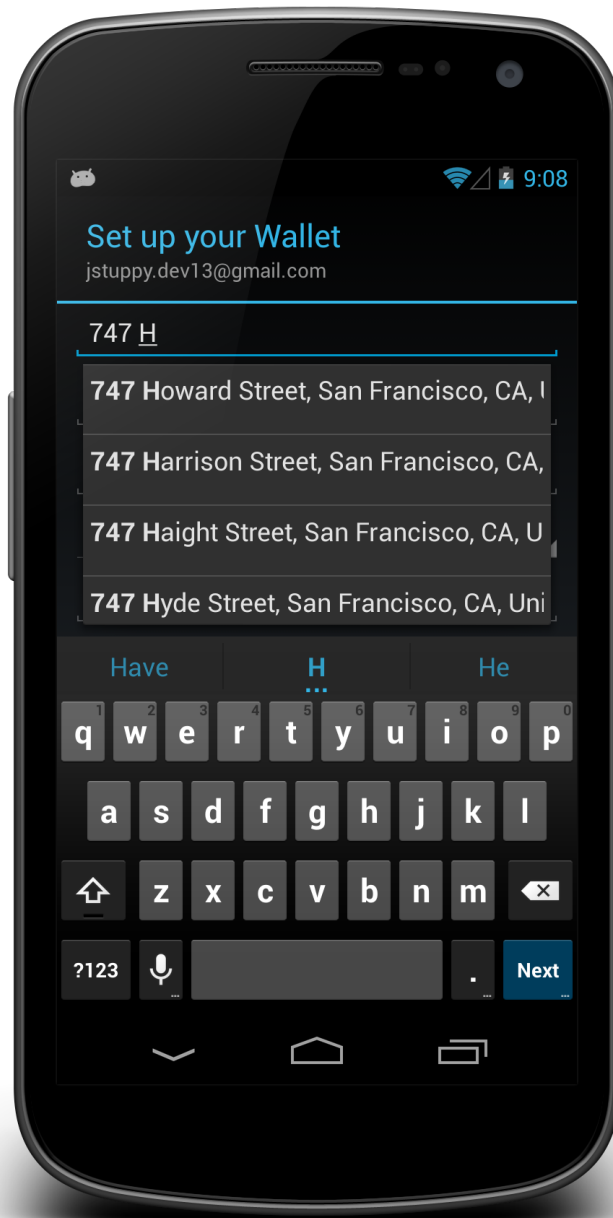
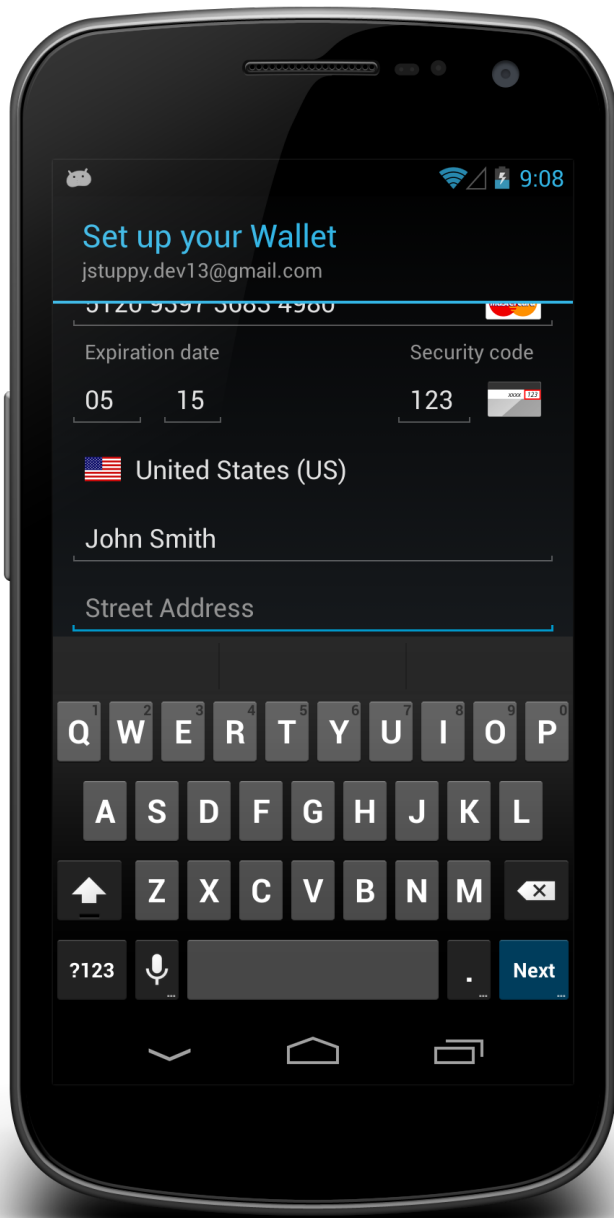
Google  
Developers

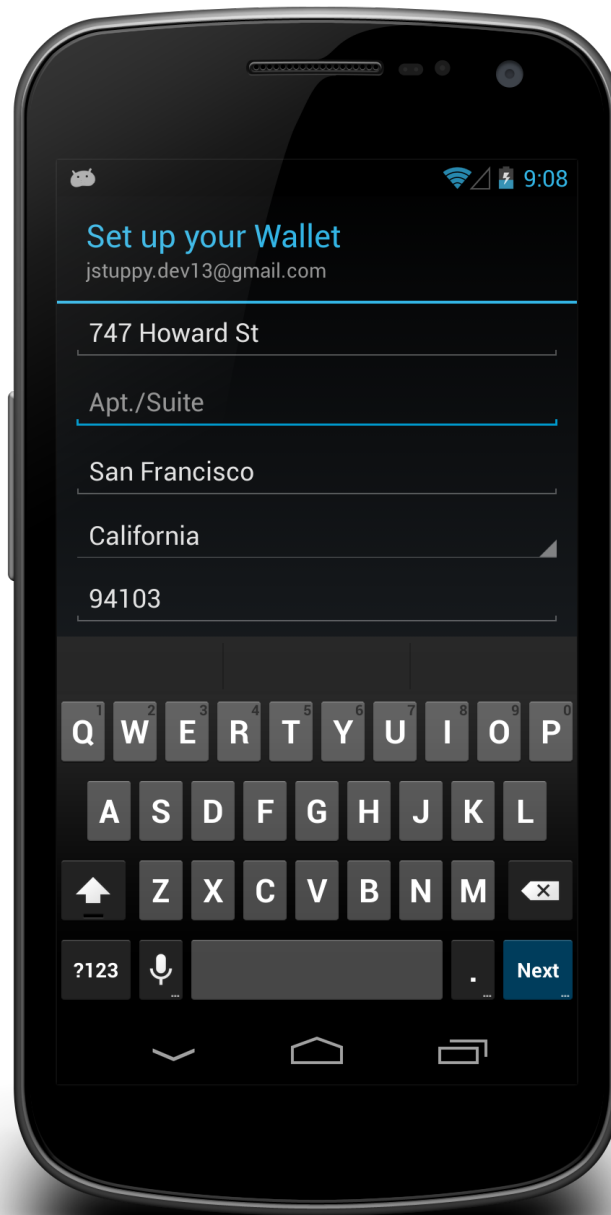


# Mobile Friendly Forms Demo Backup











# Fancy Demo Backup



