

Stealing the Web's Best Feature

Business Logic

Business logic

Business logic

Business logic

- What is a valid username?

Business logic

- What is a valid username?
 - 4-20 characters

Business logic

- What is a valid username?
 - 4-20 characters
 - [a-zA-Z0-9_\-]

Business logic

- Is your password secure enough?

Business logic

- Is your password secure enough?
 - 8+ characters

Business logic

- Is your password secure enough?
 - 8+ characters
 - 1+ lowercase

Business logic

- Is your password secure enough?
 - 8+ characters
 - 1+ lowercase
 - 1+ uppercase

Business logic

- Is your password secure enough?
 - 8+ characters
 - 1+ lowercase
 - 1+ uppercase
 - 1+ number

Business logic

- Is your password secure enough?
 - 8+ characters
 - 1+ lowercase
 - 1+ uppercase
 - 1+ number
 - 1+ special character

Business logic

- When do you nag for a rating?

Business logic

- When do you nag for a rating?
 - After 1 week?

Business logic

- When do you nag for a rating?
 - After 1 week?
 - After 3 successful payments?

Business logic

- Maximum length of a tweet?

Business logic

- Maximum length of a tweet?
 - 140 characters

Business logic

- Maximum length of a tweet?
 - 140...ish characters

Business logic

- How do you render your data?

Business logic

- How do you render your data?
 - Status text is green/red/grey

Business logic

- How do you render your data?
 - Status text is green/red/grey
 - Show appropriate icon

Business logic

- How do you render your data?
 - Status text is green/red/grey
 - Show appropriate icon
 - enum → String

Business logic

- How does your referral program work?

Business logic

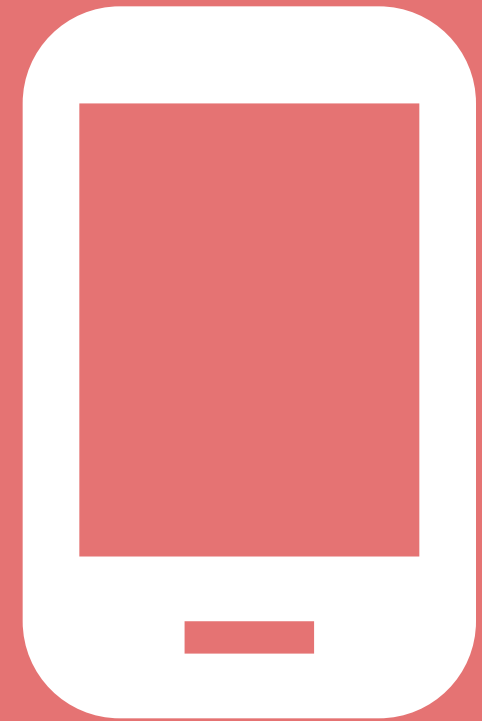
- How does your referral program work?
 - \$1 per referral?

Business logic

- How does your referral program work?
 - \$1 per referral?
 - \$10 after referring 5 people?

Where should business logic live?

Where should business logic live?

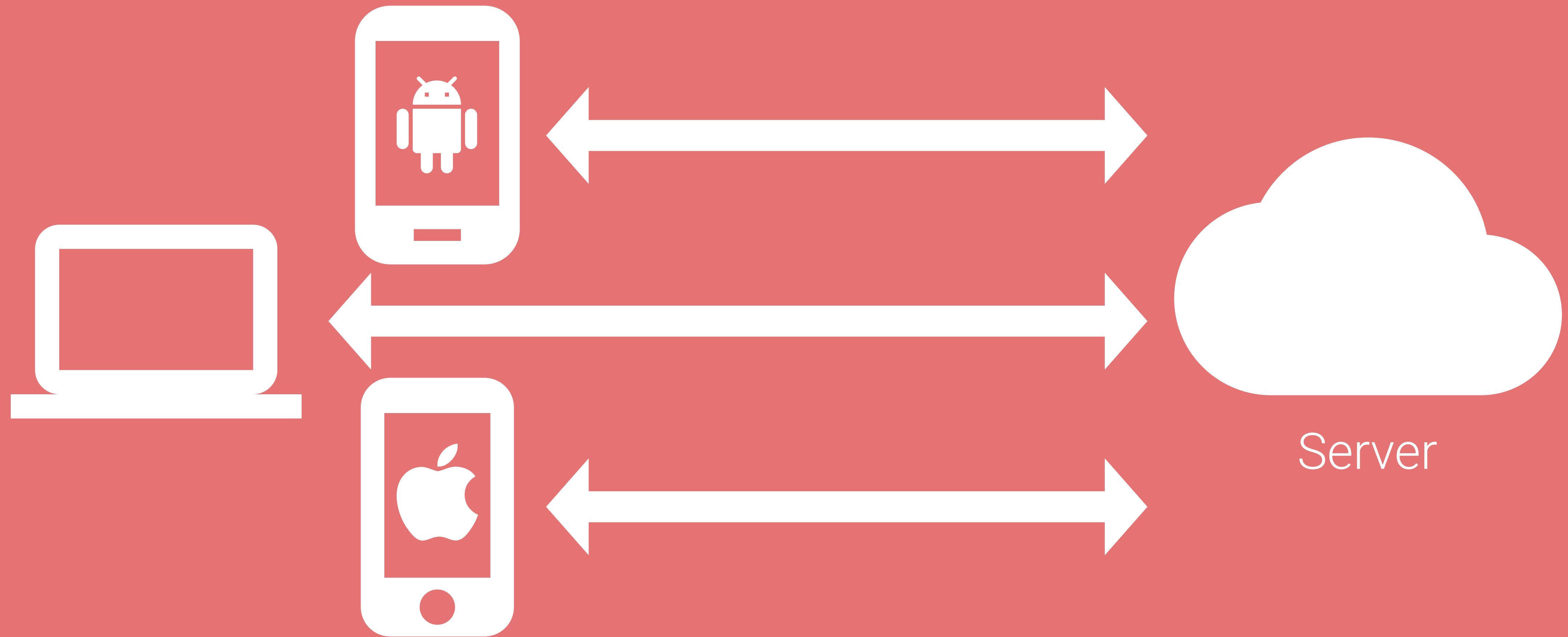


Client

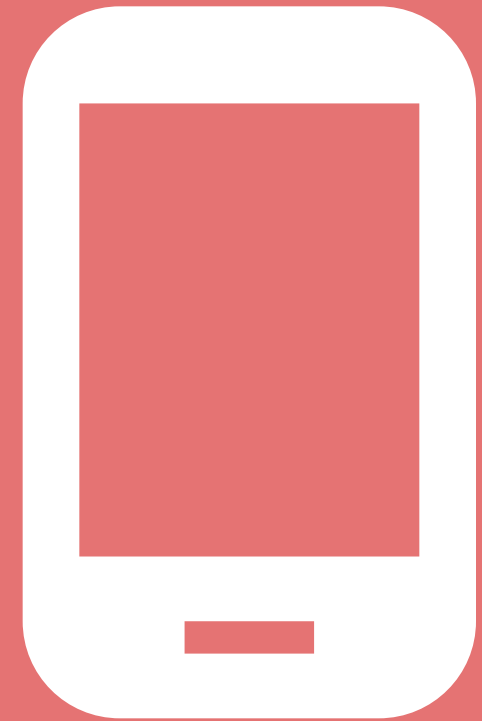


Server

Where should business logic live?



Where should business logic live?

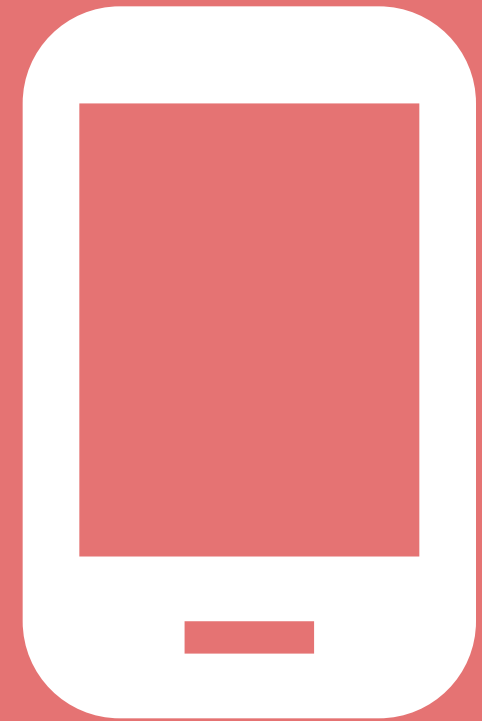


Client



Server

Where should business logic live?



Client



Server

Client

Client

- Bake the logic into your APK

Client

- Bake the logic into your APK
- Logic is fixed for a given app version

Client

+ Fast feedback loop

Client

- + Fast feedback loop
- Requires an app update to change

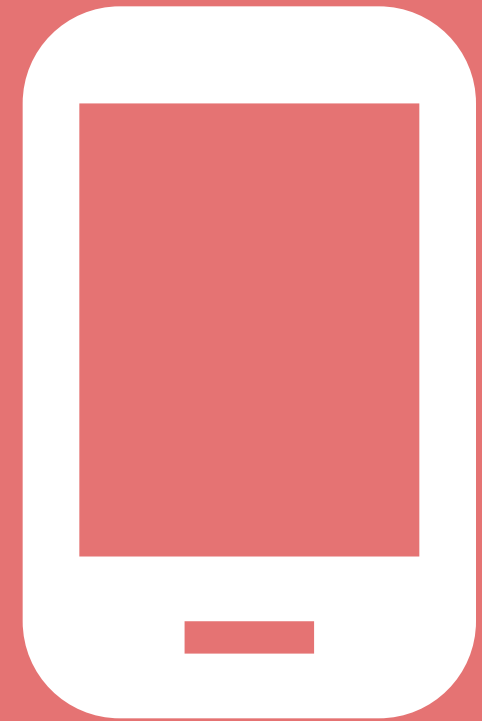
Client

- + Fast feedback loop
- Requires an app update to change
- Duplicated across all platforms

Client

- + Fast feedback loop
- Requires an app update to change
- Duplicated across all platforms
- Potential mismatch between platforms

Where should business logic live?

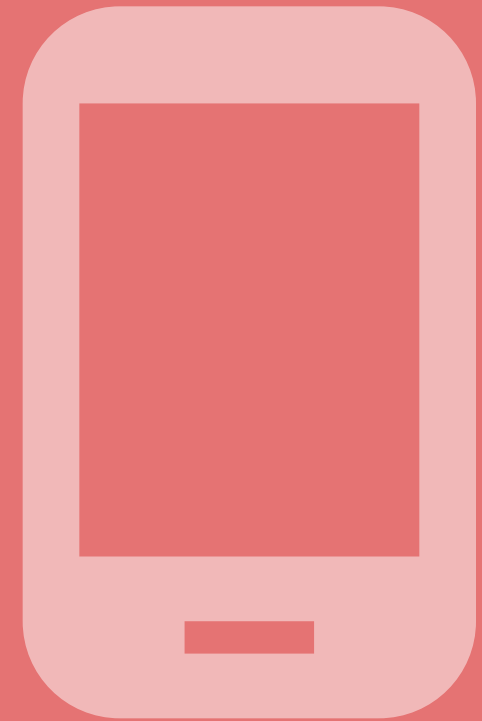


Client



Server

Where should business logic live?



Client



Server

Server

Server

- Client doesn't care

Server

- Client doesn't care
- Send the data to the server and let the server handle it

Server

- + Logic can be updated independent of app updates

Server

- + Logic can be updated independent of app updates
- + All platforms are in sync

Server

- + Logic can be updated independent of app updates
- + All platforms are in sync
- Slow feedback loop

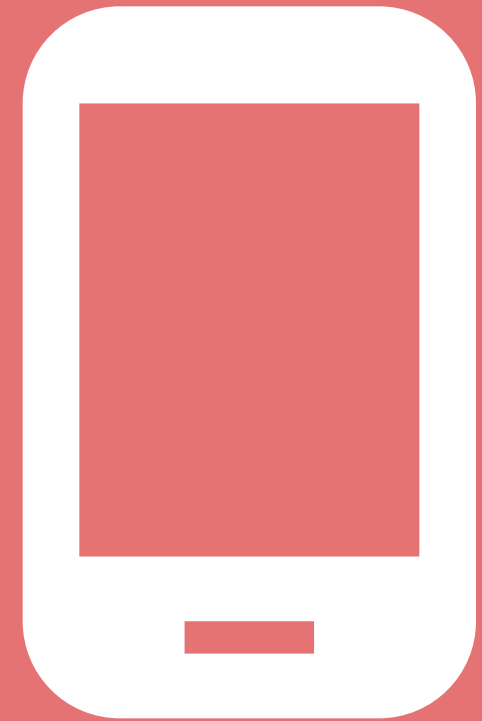
Server

- + Logic can be updated independent of app updates
- + All platforms are in sync
- Slow feedback loop
- Client must be online

Server

- + Logic can be updated independent of app updates
- + All platforms are in sync
- Slow feedback loop
- Client must be online
- Harder client complexity (new round trips)

Where should business logic live?



Client



Server

Where should business logic live?



Client



Server

Client + Server

Client + Server

Client + Server

- Client bakes in variable logic

Client + Server

- Client bakes in variable logic
- Values are provided by the server

Client + Server

- Client bakes in variable logic
- Values are provided by the server
- Feature/experiment flags

Client + Server

+ Fast feedback loop

Client + Server

- + Fast feedback loop
- + Logic updated independent of app updates*

Client + Server

- + Fast feedback loop
- + Logic updated independent of app updates*
- + All platforms are in sync*

Client + Server

- + Fast feedback loop
- + Logic updated independent of app updates*
- + All platforms are in sync*
- Requires app update to add new rules

Client + Server

- + Fast feedback loop
- + Logic updated independent of app updates*
- + All platforms are in sync*
- Requires app update to add new rules
- Platforms can fall out of sync with new rules

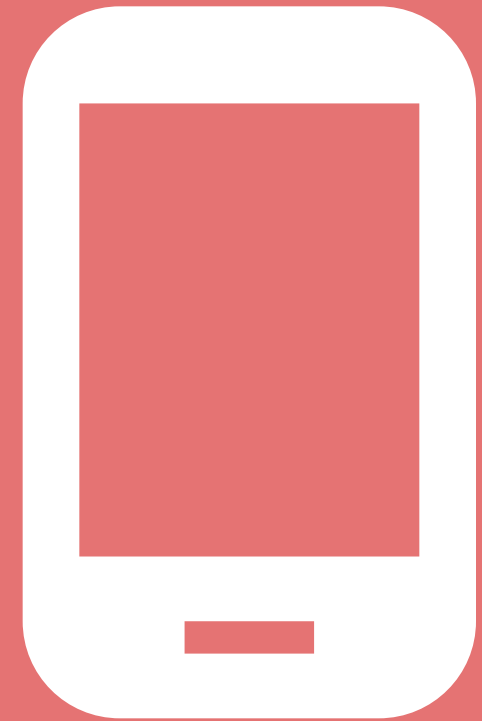
Client + Server

- + Fast feedback loop
- + Logic updated independent of app updates*
- + All platforms are in sync*
- Requires app update to add new rules
- Platforms can fall out of sync with new rules
- Duplicated across all platforms

Client + Server

- + Fast feedback loop
- + Logic updated independent of app updates*
- + All platforms are in sync*
- Requires app update to add new rules
- Platforms can fall out of sync with new rules
- Duplicated across all platforms
- Medium-hard client complexity

Where should business logic live?



Client



Server

tl;dr: If business logic lives on...

tl;dr: If business logic lives on...

Client

PM is 😞

tl;dr: If business logic lives on...

Client

PM is 😞

Server

Users are 😞

tl;dr: If business logic lives on...

Client

PM is 😞

Server

Users are 😞

Client + Server

Developers are 😞

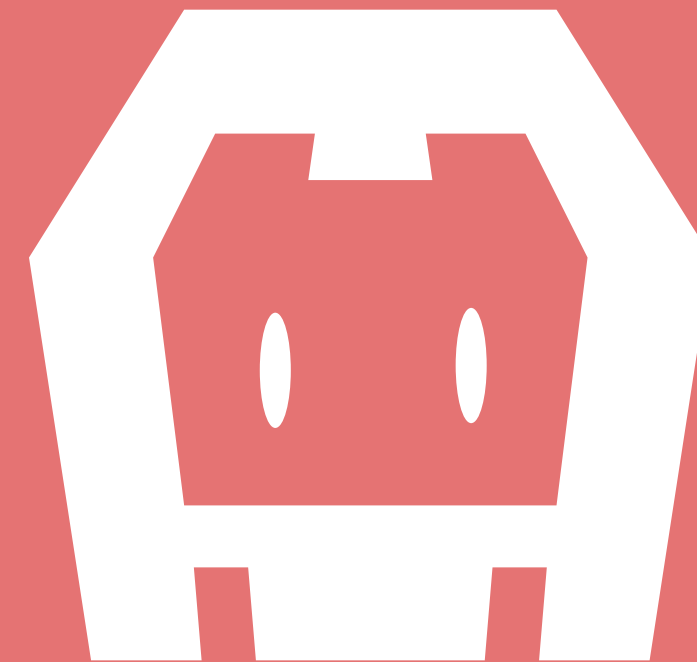
Hybrid app

Hybrid app (WebView++)

Hybrid app (WebView++)



PhoneGap



Cordova

Hybrid app (WebView++)

HTML

+ JavaScript

+ Android bits

Hybrid app (WebView++)

HTML



+ JavaScript

+ Android bits

Hybrid app (WebView++)

HTML

+ JavaScript

+ Android bits

Hybrid app (WebView++)

~~HTML~~

+ JavaScript

+ Android bits

Hybrid app (WebView++)

~~HTML~~

+ JavaScript

+ ~~Android bits~~

JavaScript

JavaScript

Web

JavaScript

Web

Yes

JavaScript

Web

iOS

Yes

JavaScript

Web

Yes

iOS

Yes

JavaScript

Web

Yes

iOS

Yes

Server

JavaScript

Web

Yes

iOS

Yes

Server

Probably

JavaScript

Web

Yes

iOS

Yes

Server

Probably

Android

JavaScript

Web

Yes

iOS

Yes

Server

Probably

Android

?

duktape-android

<https://github.com/square/duktape-android>

Username validation

```
function validateUsername(username) {  
    return username.length >= 4 && username.length <= 20;  
}
```

Username validation

```
function validateUsername(username) {  
    if (username.length < 4 || username.length > 20) {  
        return "Username must be between 4 and 20 characters.";  
    }  
  
    return null;  
}
```

Username validation

```
Duktape duktape = Duktape.create();  
duktape.evaluate(myScript, "my_script.js");
```

Username validation

```
Duktape duktape = Duktape.create();  
duktape.evaluate(myScript, "my_script.js");
```

Username validation

```
Duktape duktape = Duktape.create();  
duktape.evaluate(myScript, "my_script.js");
```


Username validation

```
String usernameErrorMessage = (String) duktape.evaluate(  
    String.format("validateUsername(\"%s\")", username));
```

Username validation

```
interface Validator {  
    String validateUsername(String username);  
}
```

Username validation

```
interface Validator {  
    String validateUsername(String username);  
}
```

```
var Validator = {  
    validateUsername: function(username) {  
        // Validate username.  
    }  
};
```

Username validation

```
String usernameErrorMessage = (String) duktape.evaluate(  
    String.format("validateUsername(\"%s\");", username));
```

Username validation

```
Validator validator = duktape.get("Validator", Validator.class);  
String usernameErrorMessage = (String) duktape.evaluate(  
    String.format("validateUsername(\"%s\");", username));
```

Username validation

```
Validator validator = duktape.get("Validator", Validator.class);  
String usernameErrorMessage = (String) duktape.evaluate(  
    String.format("validateUsername(\"%s\");", username));
```

Username validation

```
Validator validator = duktape.get("Validator", Validator.class);  
String usernameErrorMessage = (String) duktape.evaluate(  
    String.format("validateUsername(\"%s\");", username));
```

Username validation

```
Validator validator = duktape.get("Validator", Validator.class);  
String usernameErrorMessage = (String) duktape.evaluate(  
    String.format("validateUsername(\"%s\");", username));
```


Username validation

```
Validator validator = duktape.get("Validator", Validator.class);  
String usernameErrorMessage = (String) duktape.evaluate(  
    String.format("validateUsername(\"%s\");", username));
```

Username validation

```
Validator validator = duktape.get("Validator", Validator.class);  
String usernameErrorMessage =  
    validator.validateUsername(username);
```

Username validation

```
Validator validator = duktape.get("Validator", Validator.class);  
String usernameErrorMessage =  
    validator.validateUsername(username);
```

Java from JS

Java from JS: Logging

```
interface JsLogger {  
    void d(String message);  
}
```

Java from JS: Logging

```
JsLogger jsLogger = new JsLogger() {  
    @Override public void d(String message) {  
        Timber.d(message);  
    }  
};
```

Java from JS: Logging

```
JsLogger jsLogger = new JsLogger() {  
    @Override public void d(String message) {  
        Timber.d(message);  
    }  
};
```

```
duktape.set("JsLogger", JsLogger.class, jsLogger);
```

Java from JS: Logging

```
JsLogger jsLogger = new JsLogger() {  
    @Override public void d(String message) {  
        Timber.d(message);  
    }  
};
```

```
duktape.set("JsLogger", JsLogger.class, jsLogger);
```


Java from JS: Logging

```
JsLogger jsLogger = new JsLogger() {  
    @Override public void d(String message) {  
        Timber.d(message);  
    }  
};
```

```
duktape.set("JsLogger", JsLogger.class, jsLogger);
```

Java from JS: Logging

```
JsLogger jsLogger = new JsLogger() {  
    @Override public void d(String message) {  
        Timber.d(message);  
    }  
};
```

```
duktape.set("JsLogger", JsLogger.class, jsLogger);
```

Java from JS: Logging

```
JsLogger.d("Username length > 20");
```

Java from JS: Date formatting

```
interface DateFormatter {  
    String format(long millis);  
}
```

Java from JS: Date formatting

```
DateFormatter dateFormatter = new DateFormatter() {  
    @Override String format(long millis) {  
        return DateUtils.formatDateTime(context, millis,  
            FORMAT_SHOW_DATE | FORMAT_SHOW_YEAR);  
    }  
};
```

Java from JS: Date formatting

```
DateFormatter dateFormatter = new DateFormatter() {  
    @Override String format(long millis) {  
        return DateUtils.formatDateTime(context, millis,  
            FORMAT_SHOW_DATE | FORMAT_SHOW_YEAR);  
    }  
};
```

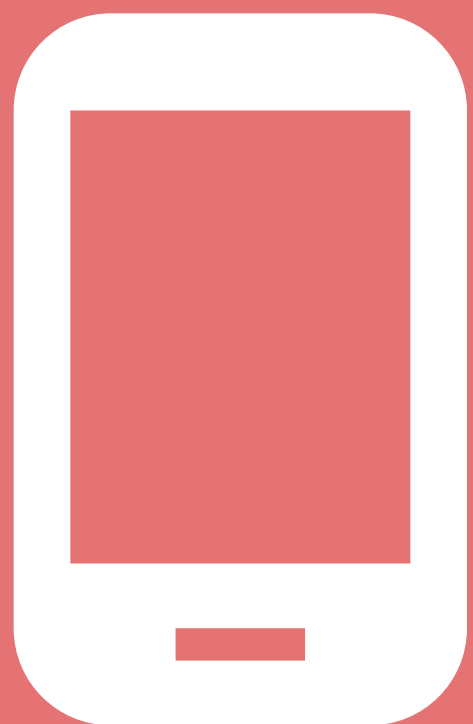
```
duktape.set("DateFormatter", DateFormatter.class,  
    dateFormatter);
```

Java from JS: Date formatting

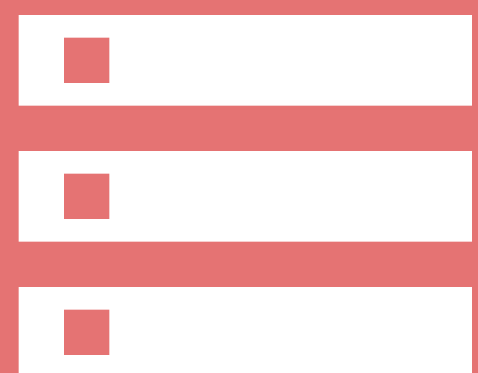
```
DateFormatter.format(Date.now());
```

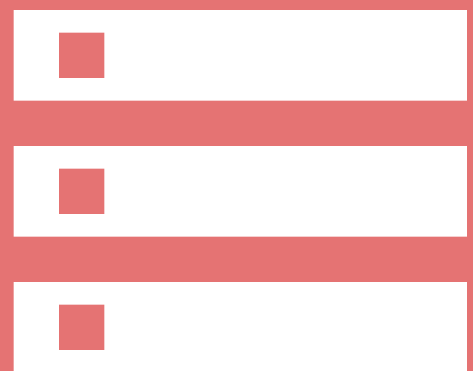
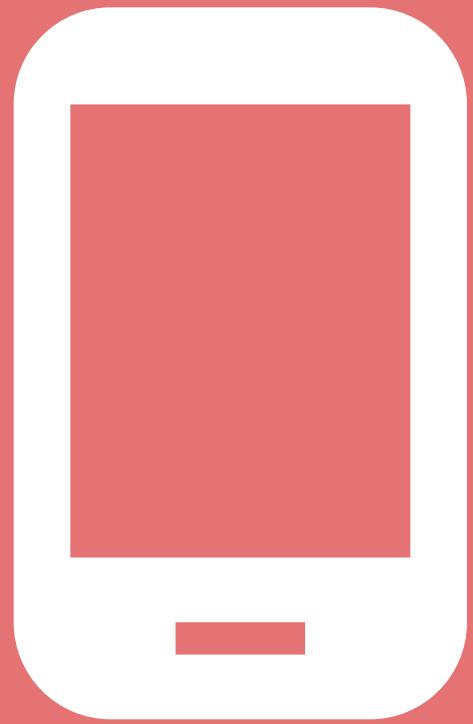
Java from JS: Storage

```
interface Storage {  
    void write(String key, String value);  
    String read(String key);  
}
```

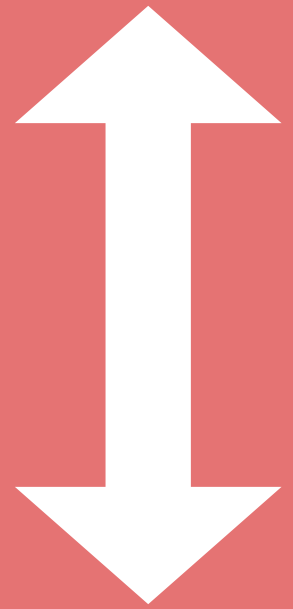













Gradle download task

```
import de.undercouch.gradle.tasks.download.Download

task updateScript(type: Download) {
    src 'https://myserver.com/my-script.js'
    dest file('src/main/res/raw/my_script.js')
}
```

JavaScript

JavaScript

+ Fast feedback loop

JavaScript

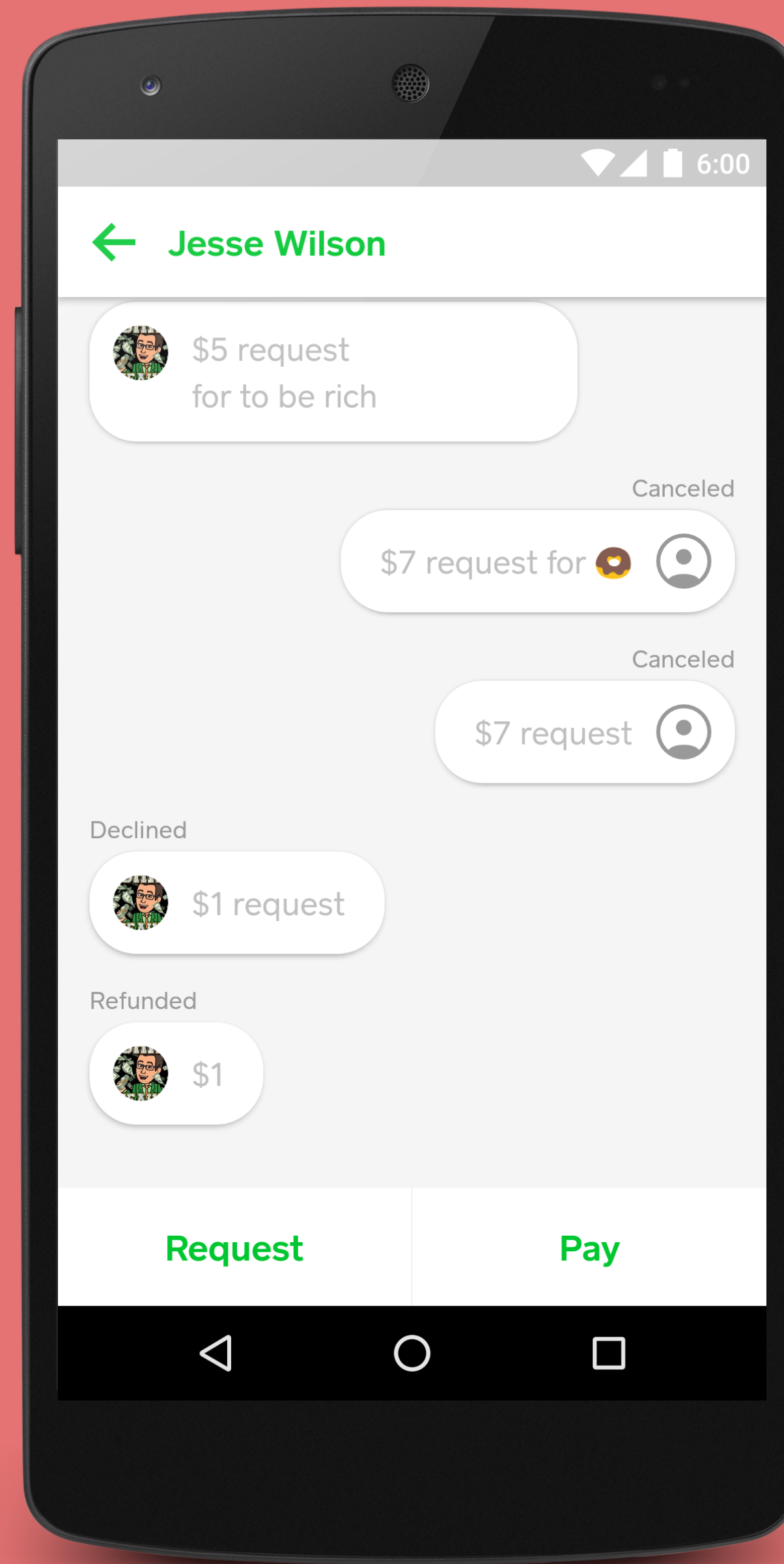
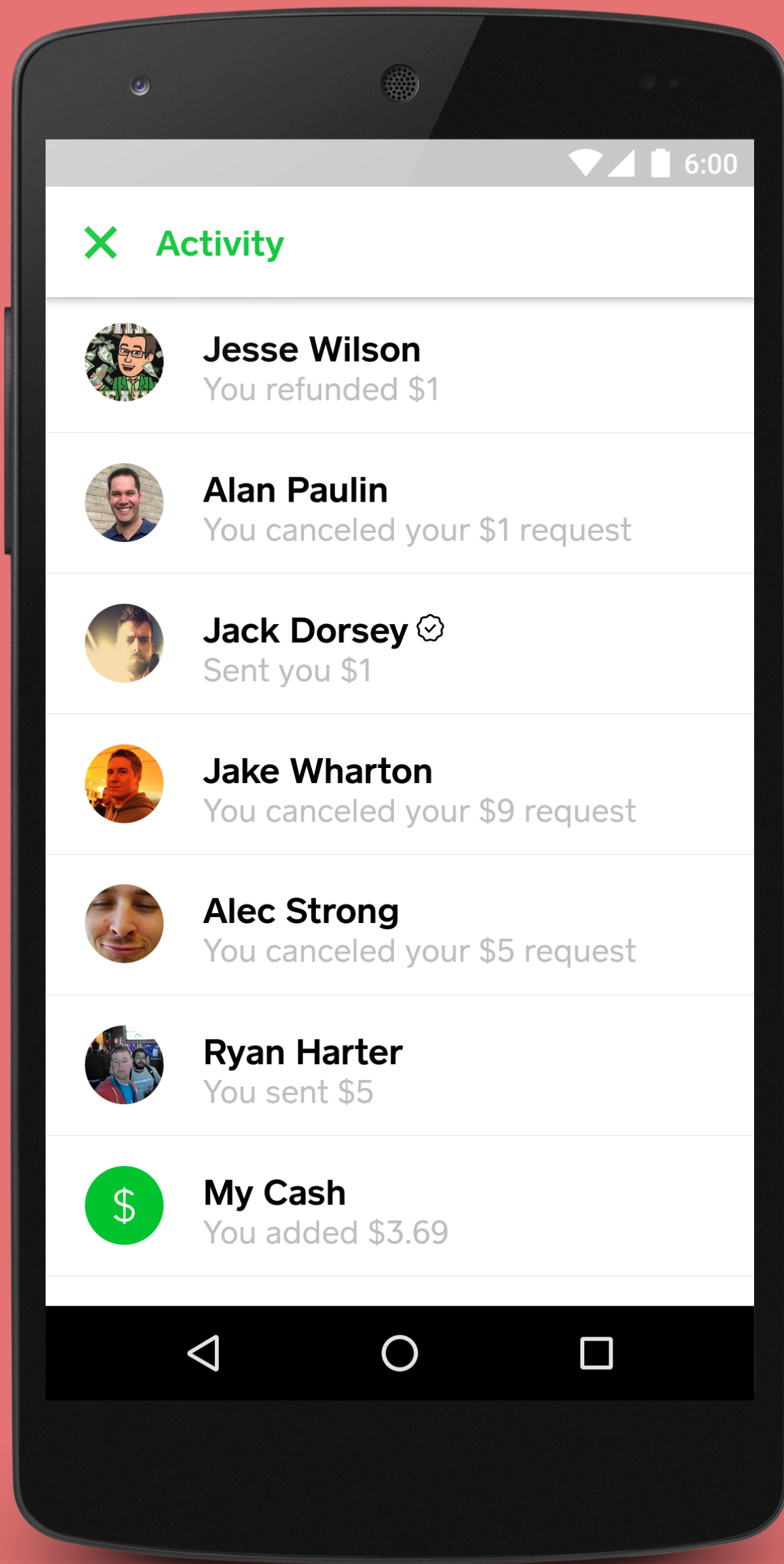
- + Fast feedback loop
- + Logic can be updated independent of app updates

JavaScript

- + Fast feedback loop
- + Logic can be updated independent of app updates
- + All platforms are in sync

JavaScript

- + Fast feedback loop
- + Logic can be updated independent of app updates
- + All platforms are in sync
- Medium client complexity for first integration (easy for subsequent)



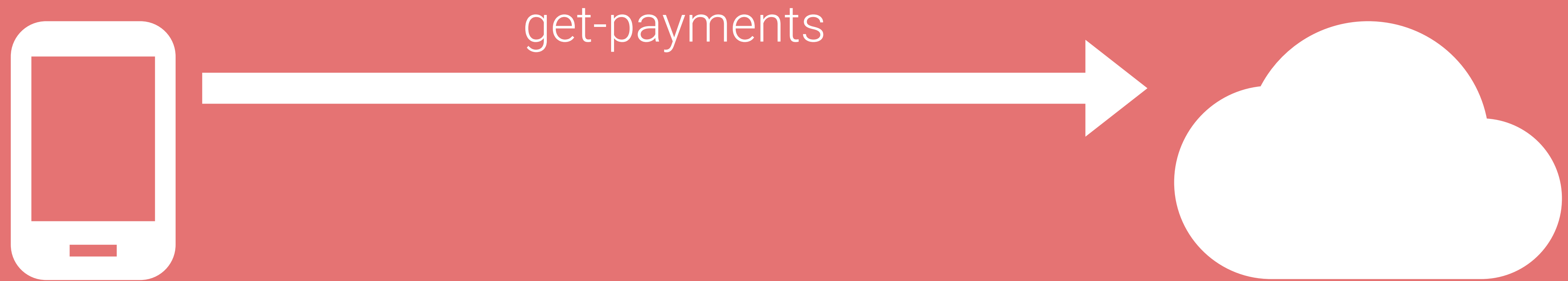
Square Cash: Activity v1

Square Cash: Activity v1

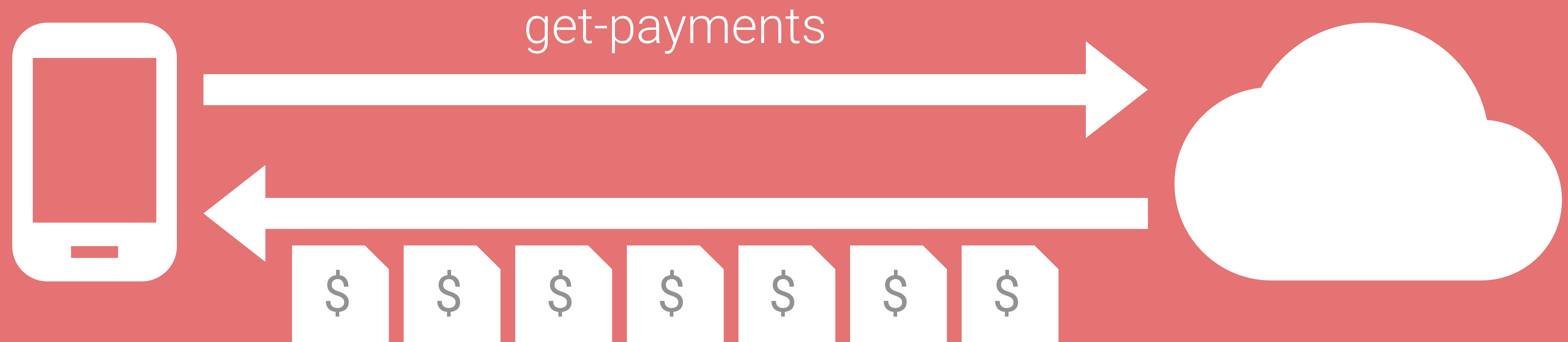
Square Cash: Activity v1



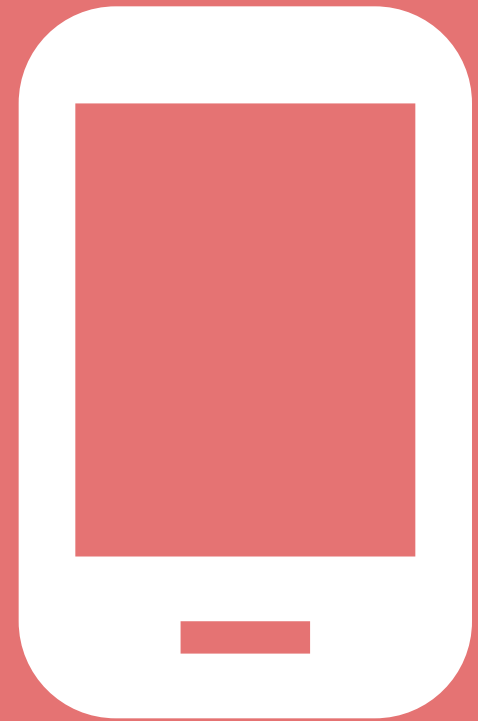
Square Cash: Activity v1



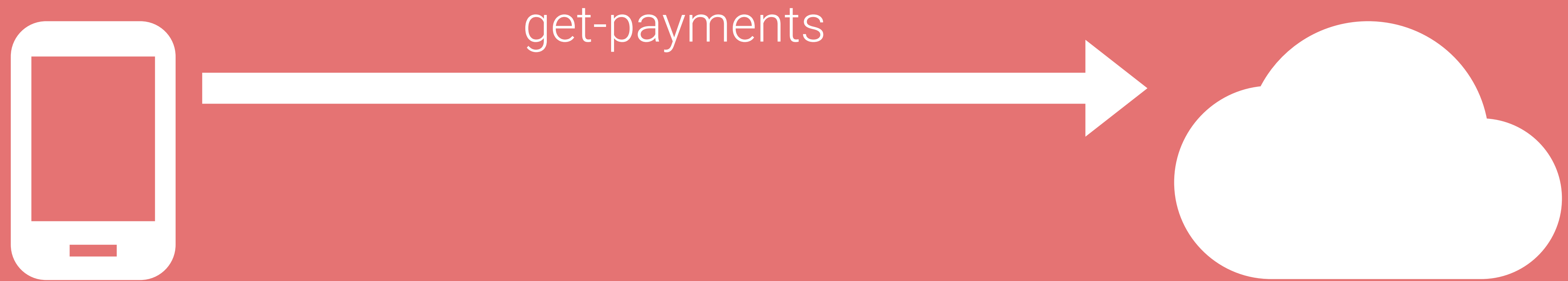
Square Cash: Activity v1



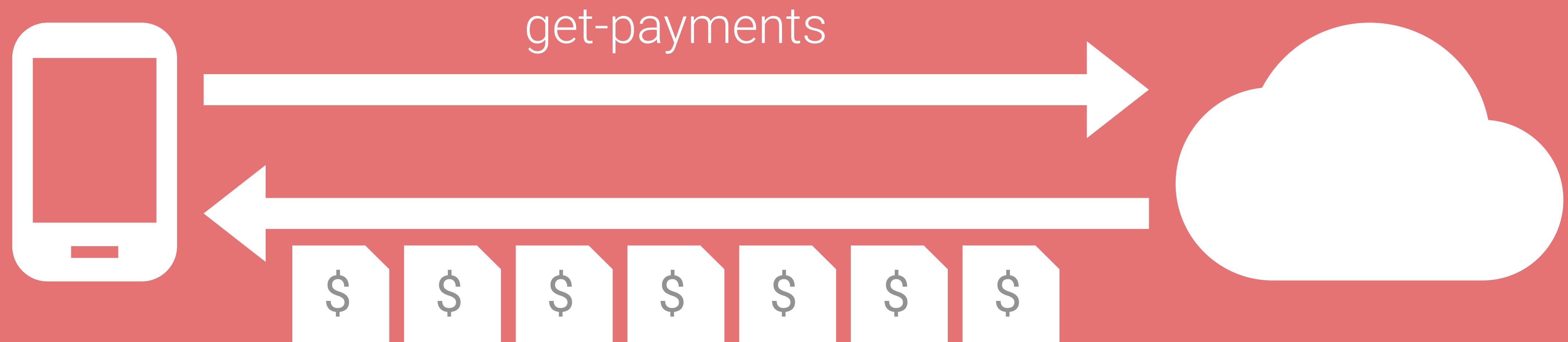
Square Cash: Activity v1



Square Cash: Activity v1



Square Cash: Activity v1



Square Cash: Activity v1



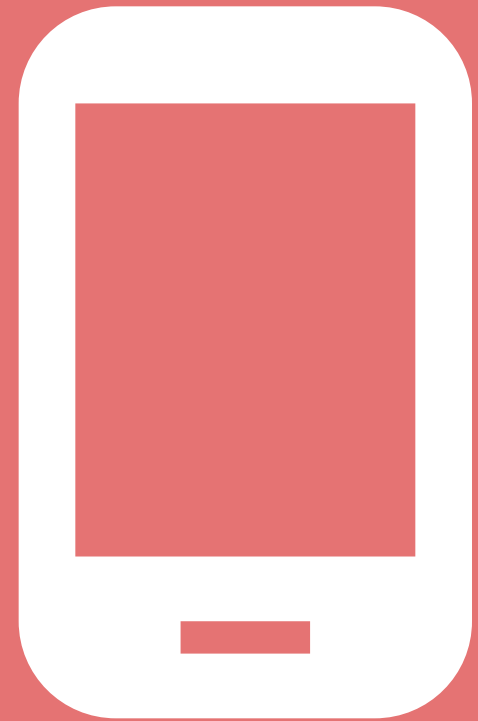
Square Cash: Activity v1

Square Cash: Activity v1

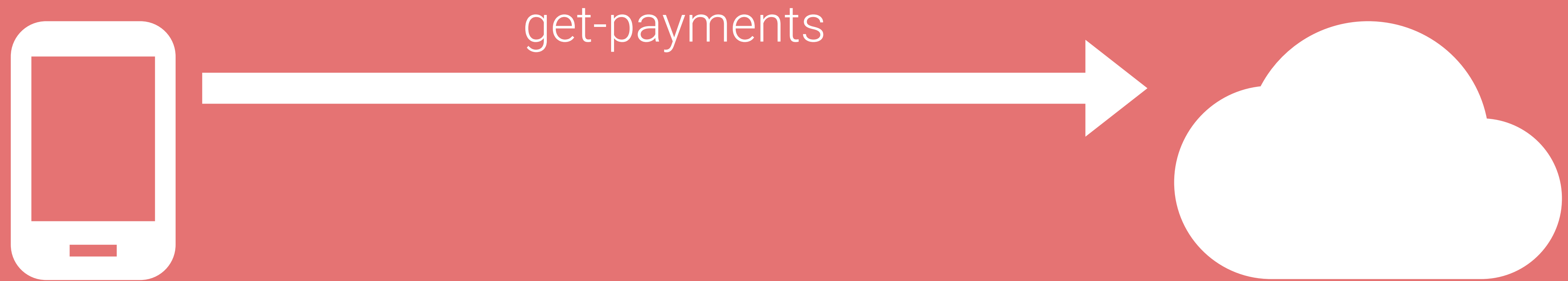
Square Cash: Activity v2

Square Cash: Activity v2

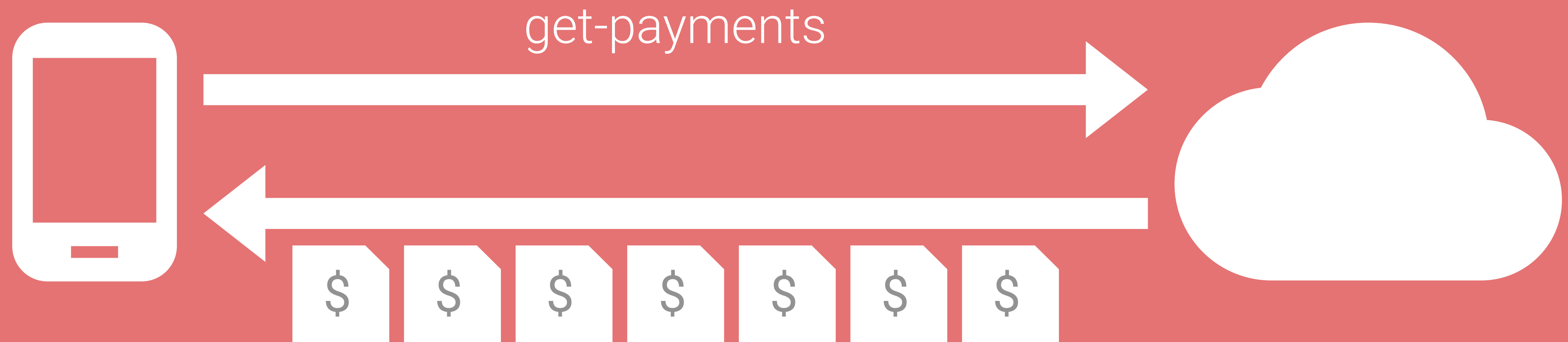
Square Cash: Activity v2



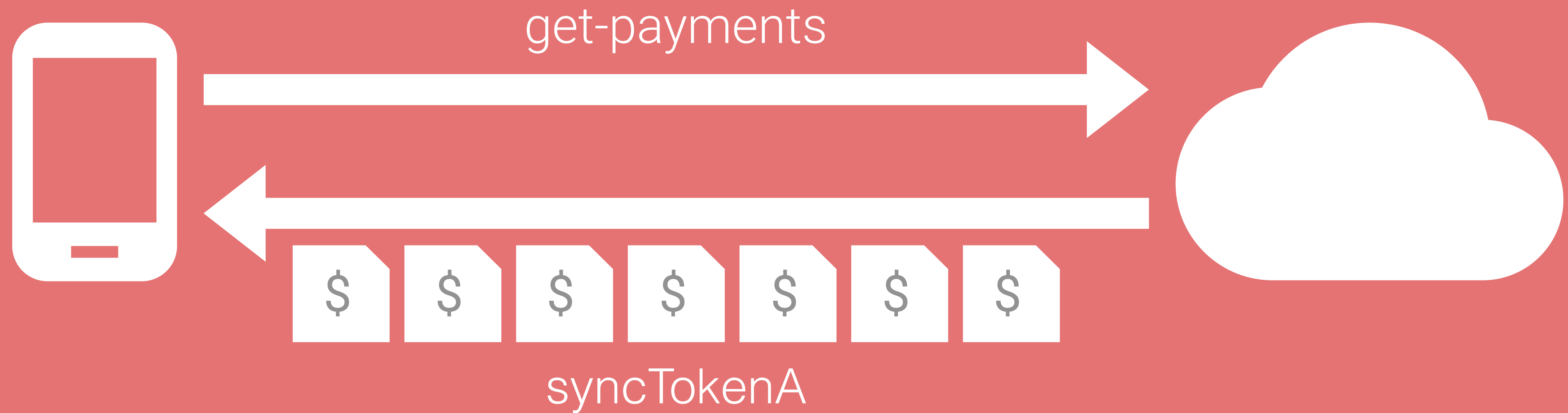
Square Cash: Activity v2



Square Cash: Activity v2



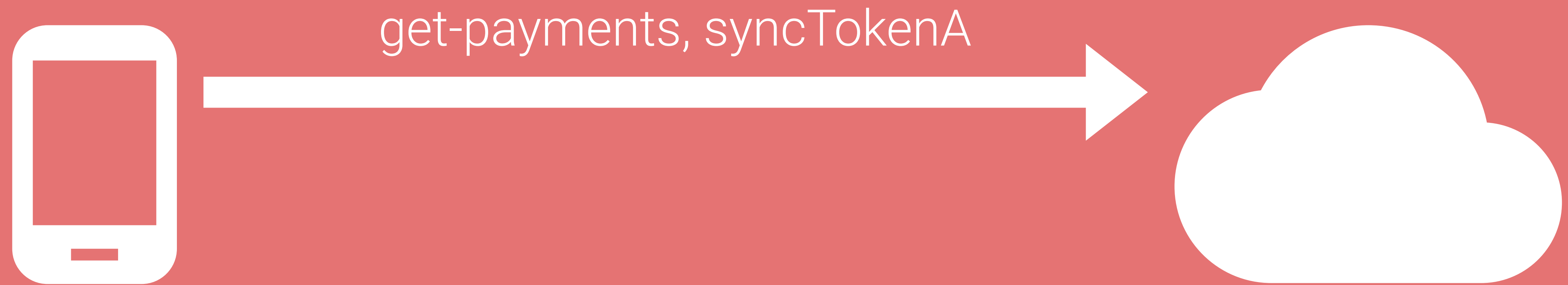
Square Cash: Activity v2



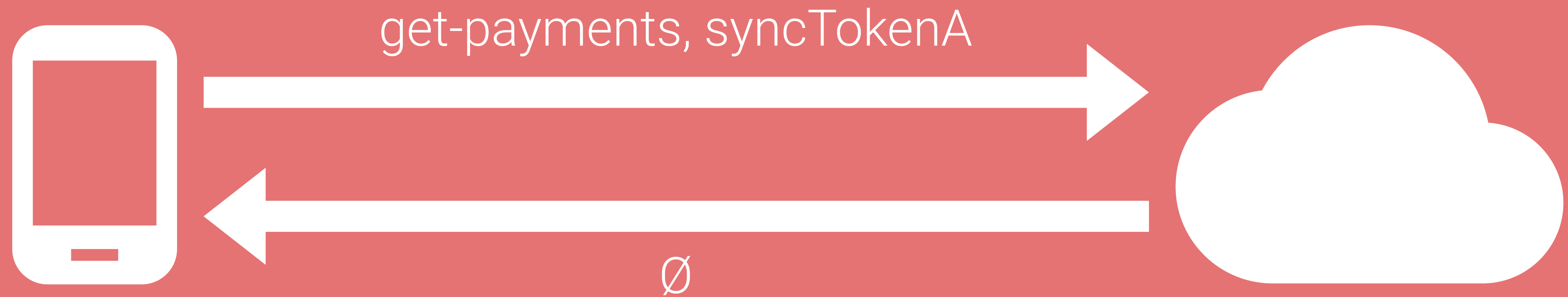
Square Cash: Activity v2



Square Cash: Activity v2



Square Cash: Activity v2



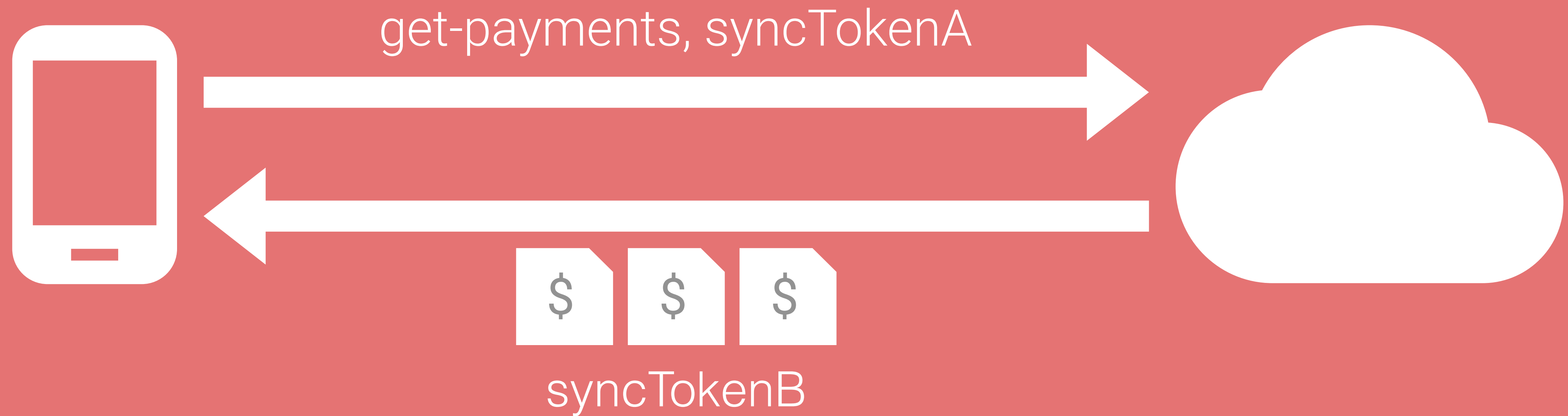
Square Cash: Activity v2

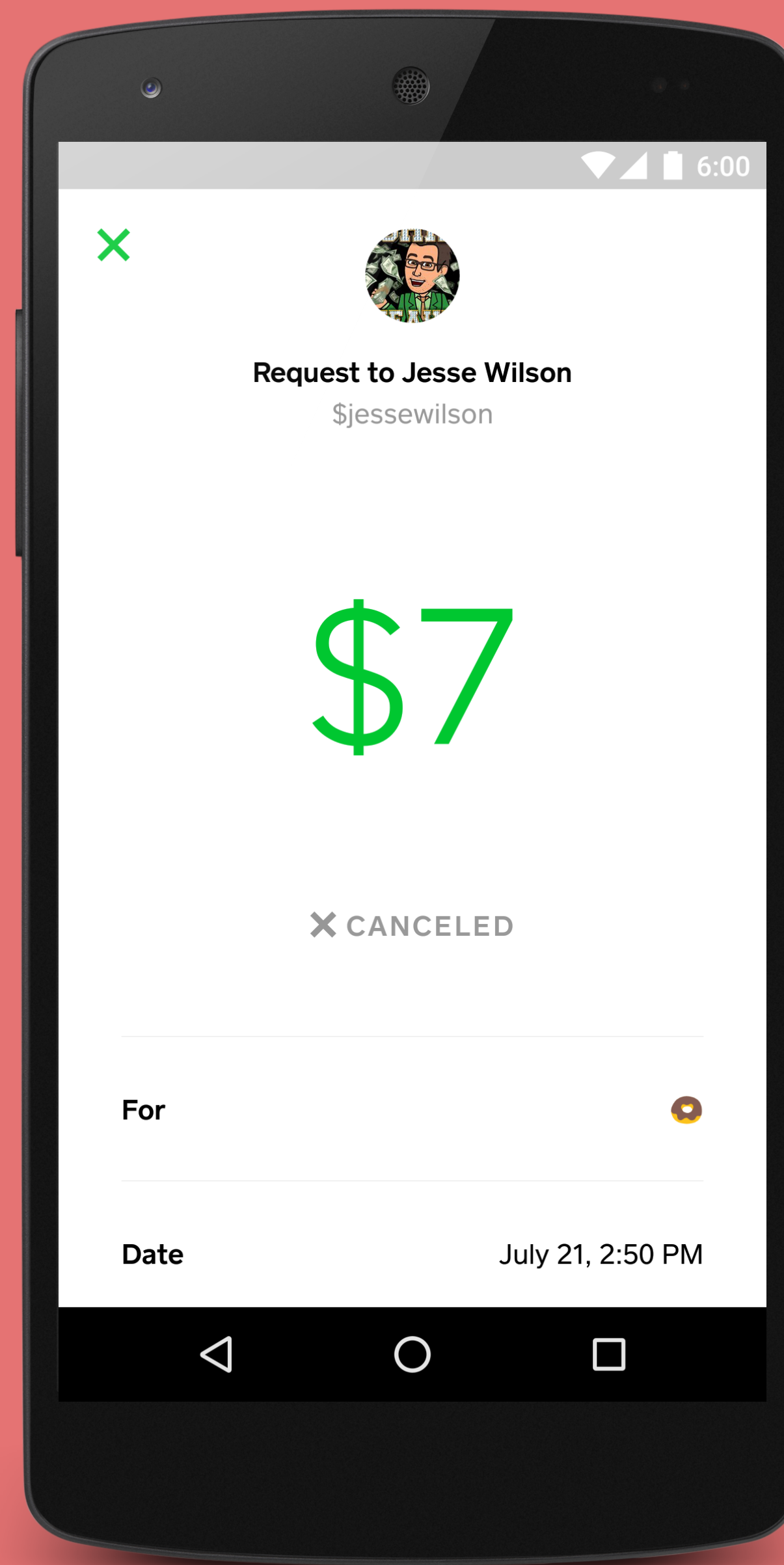
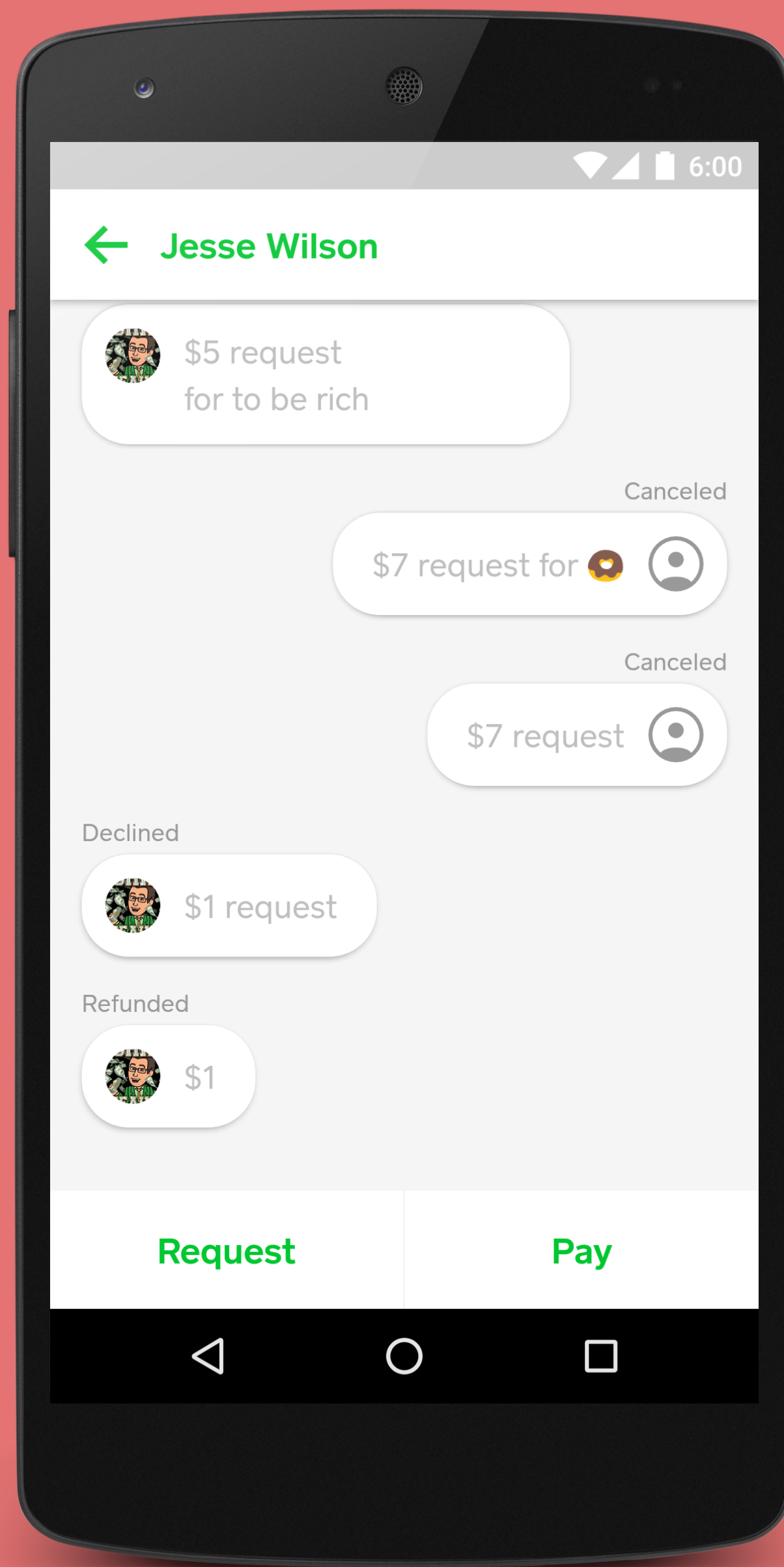
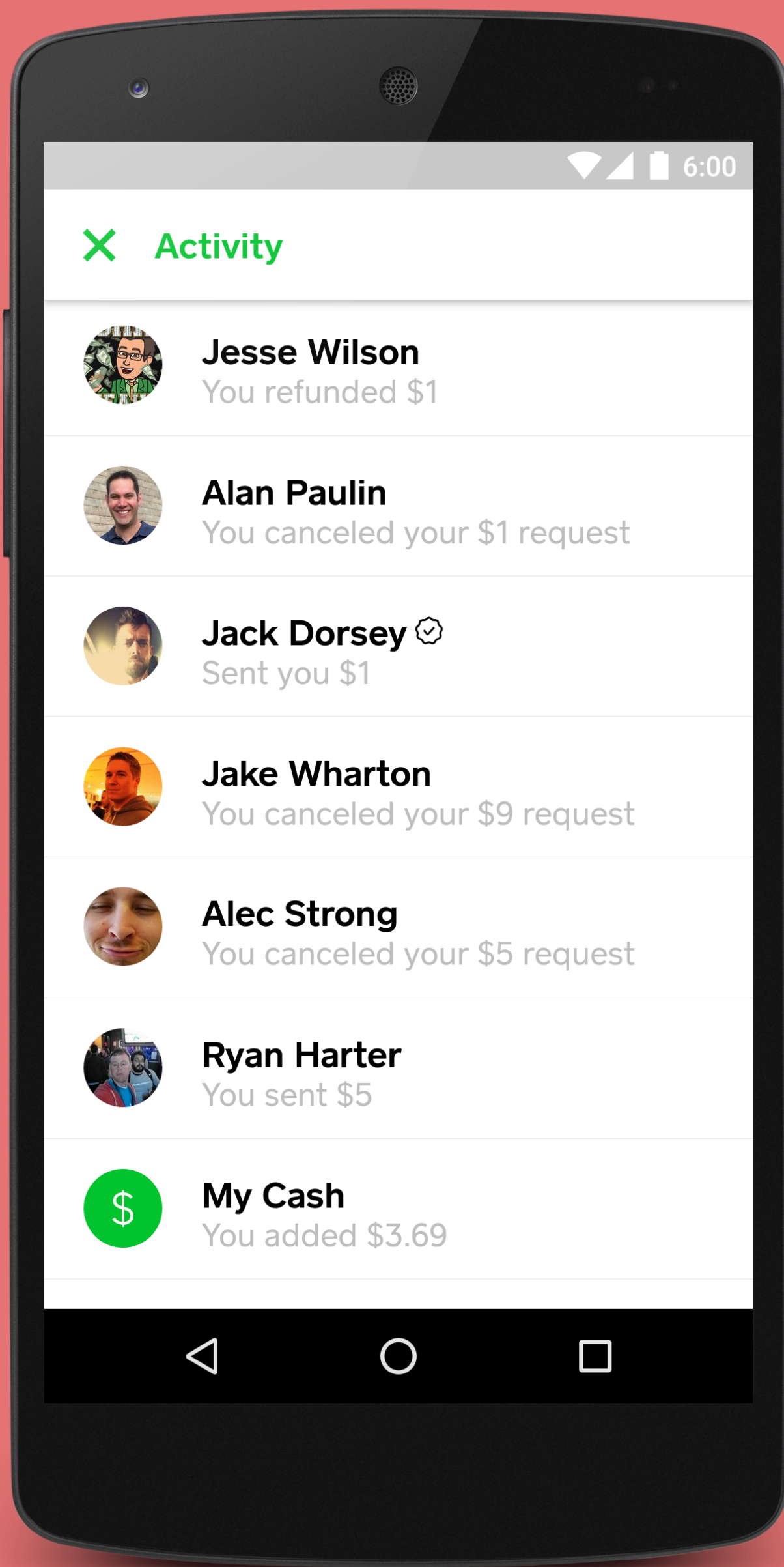


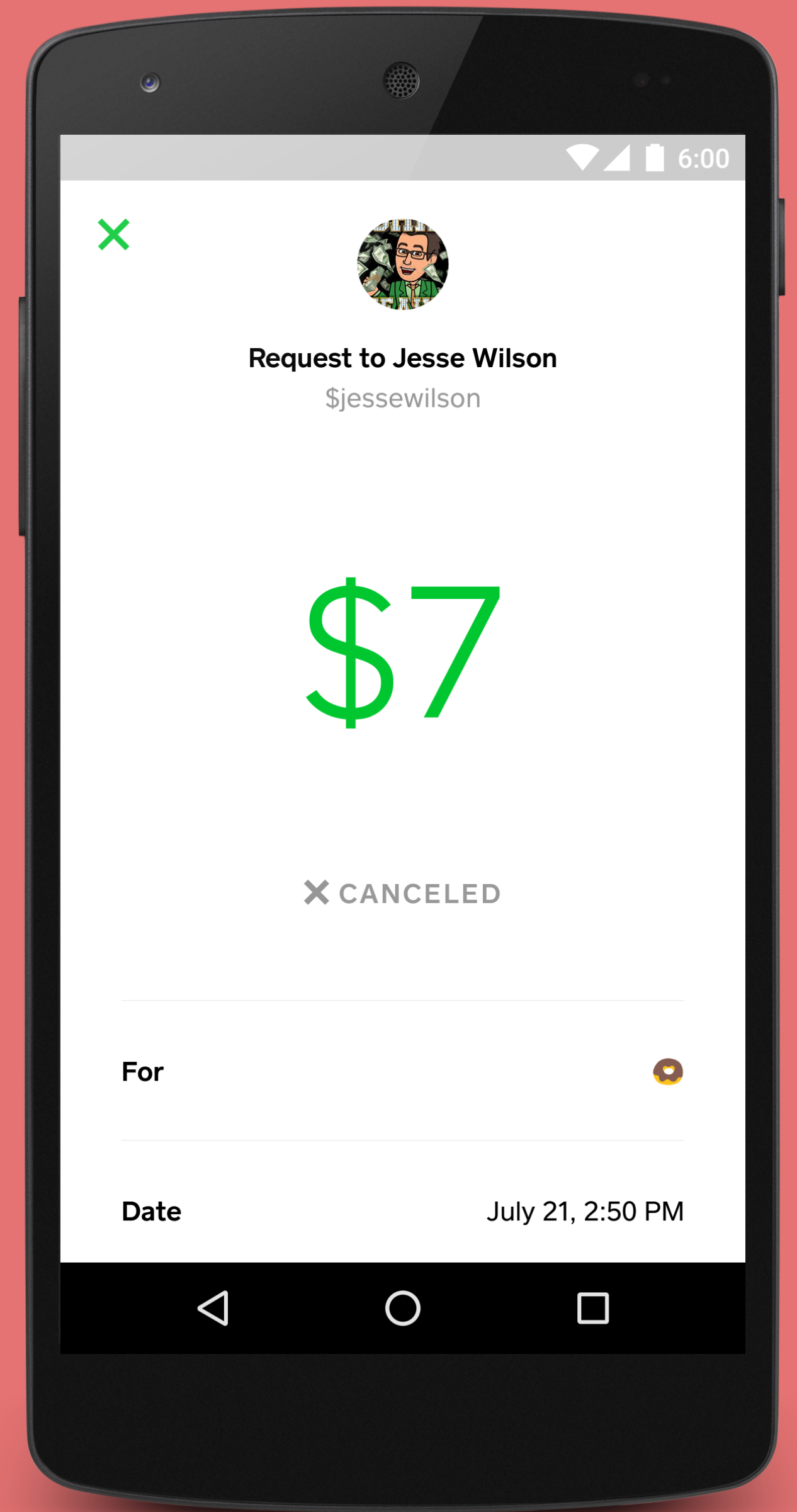
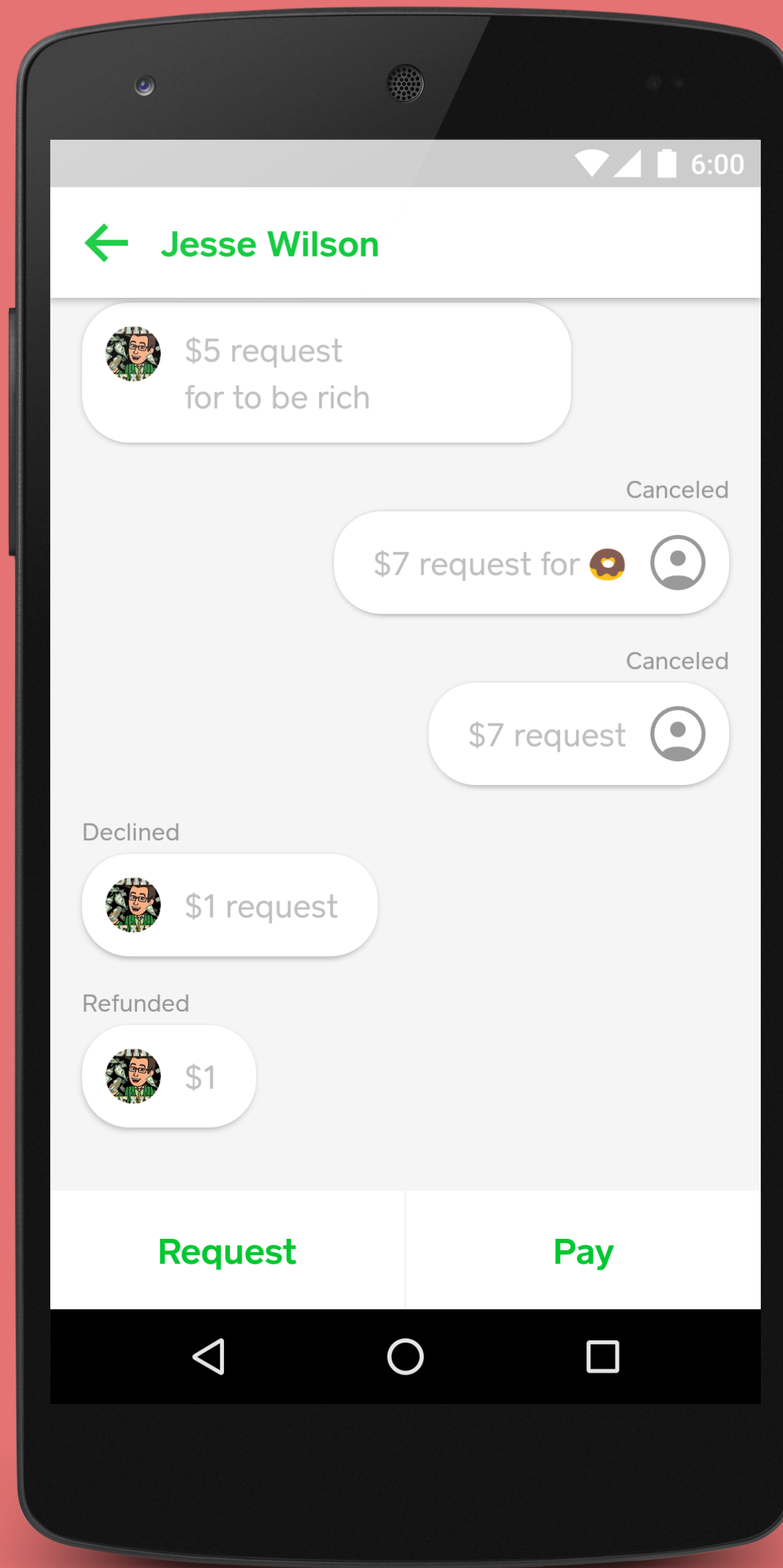
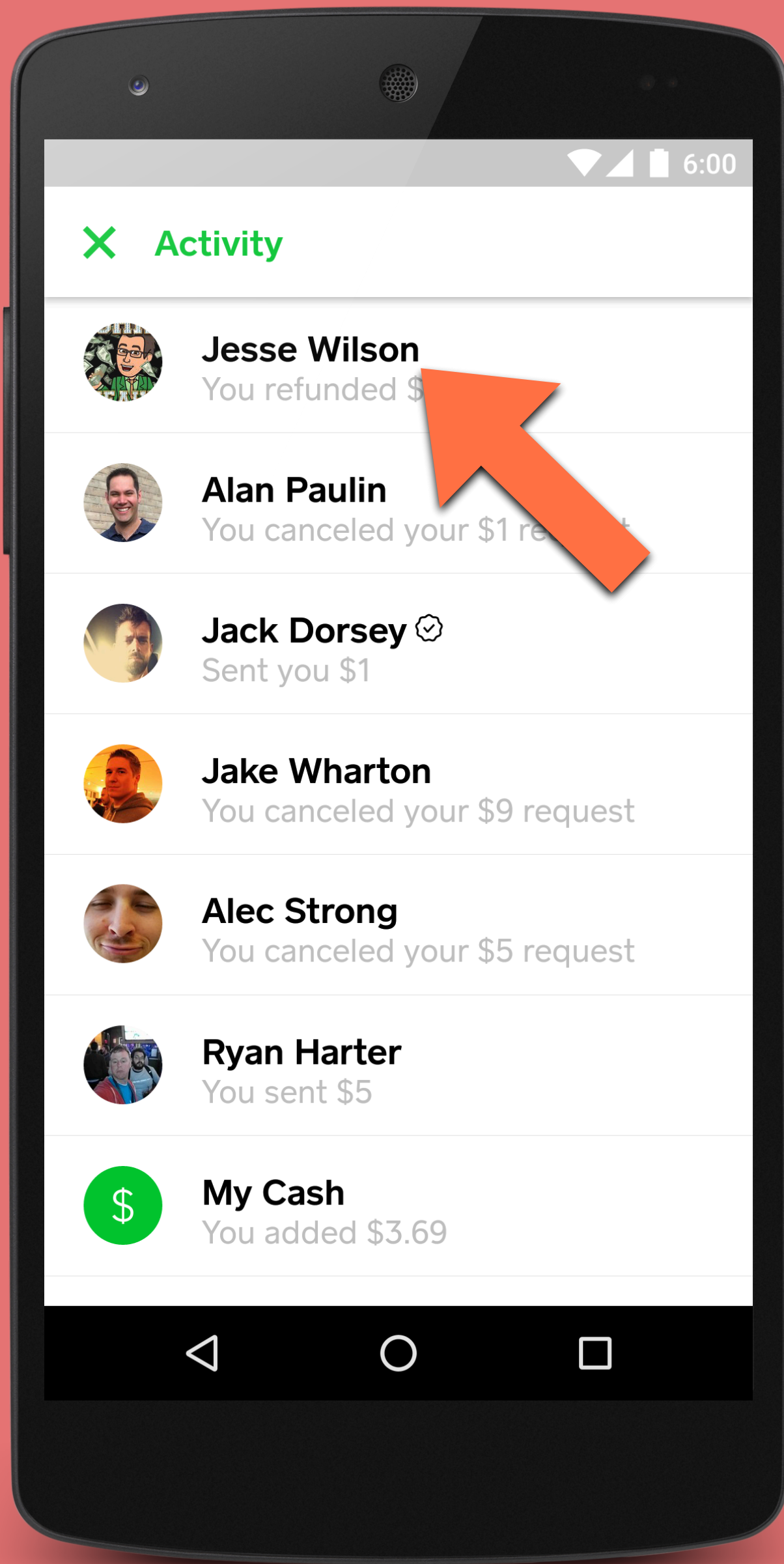
Square Cash: Activity v2

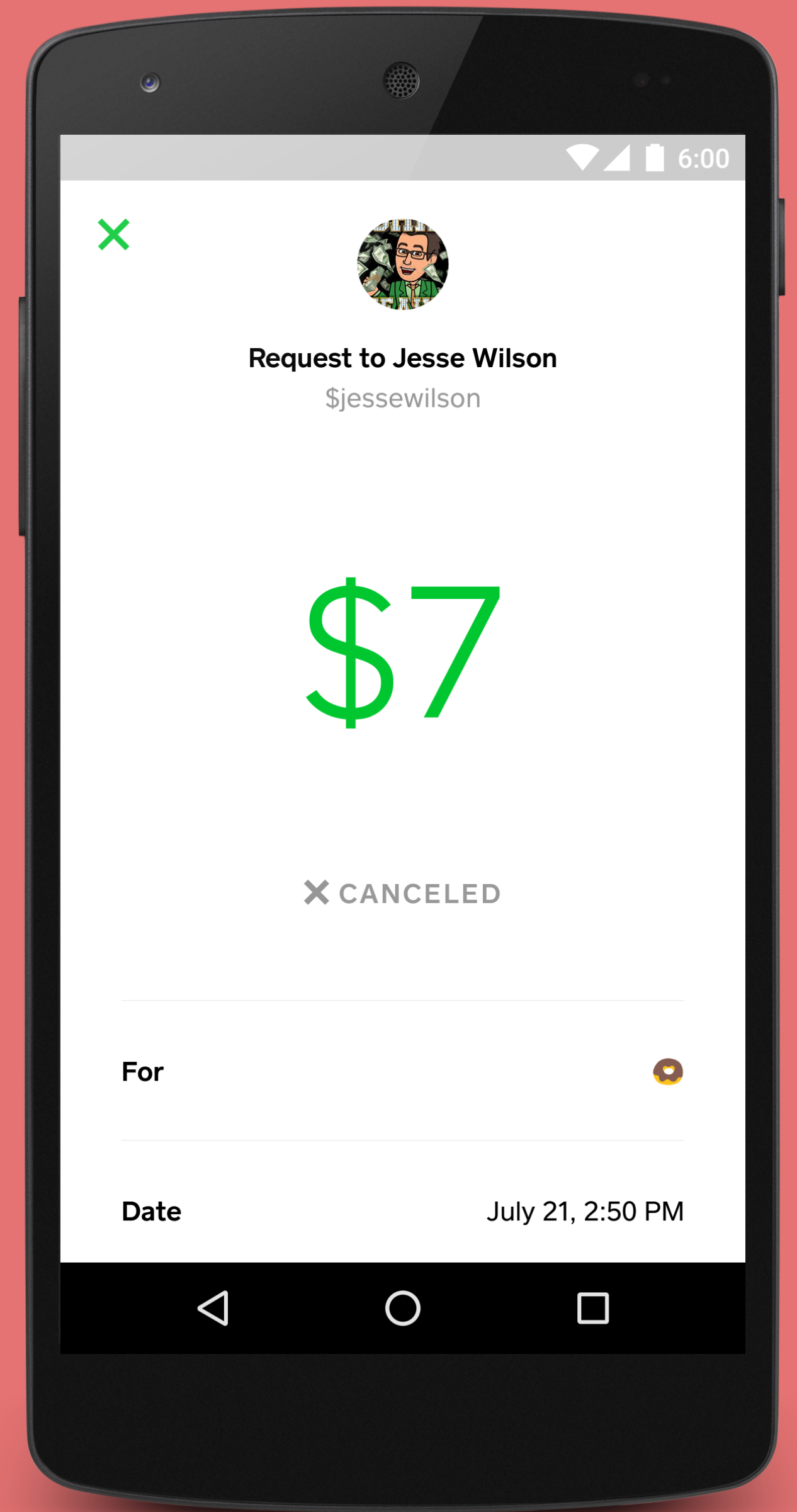
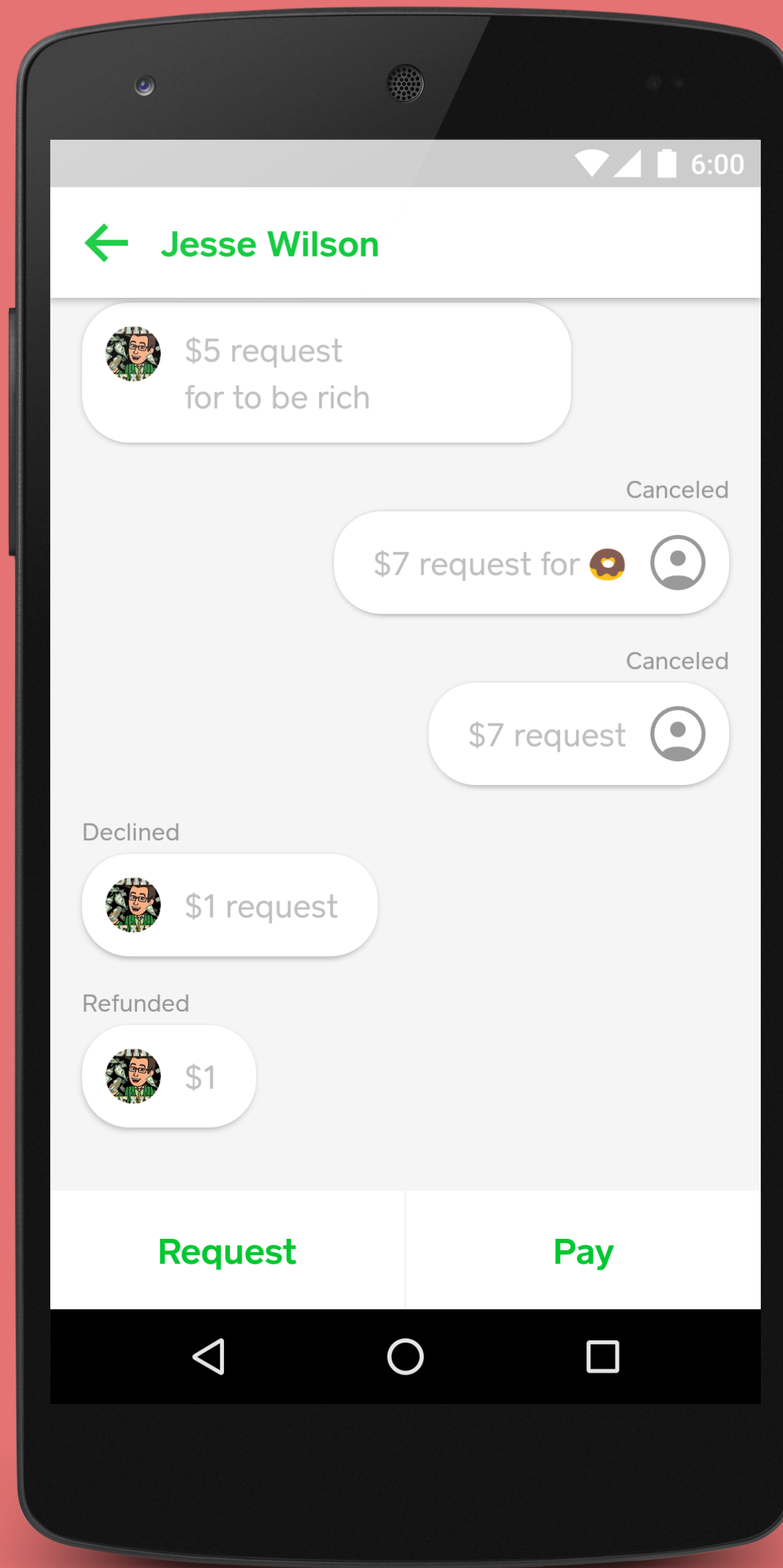
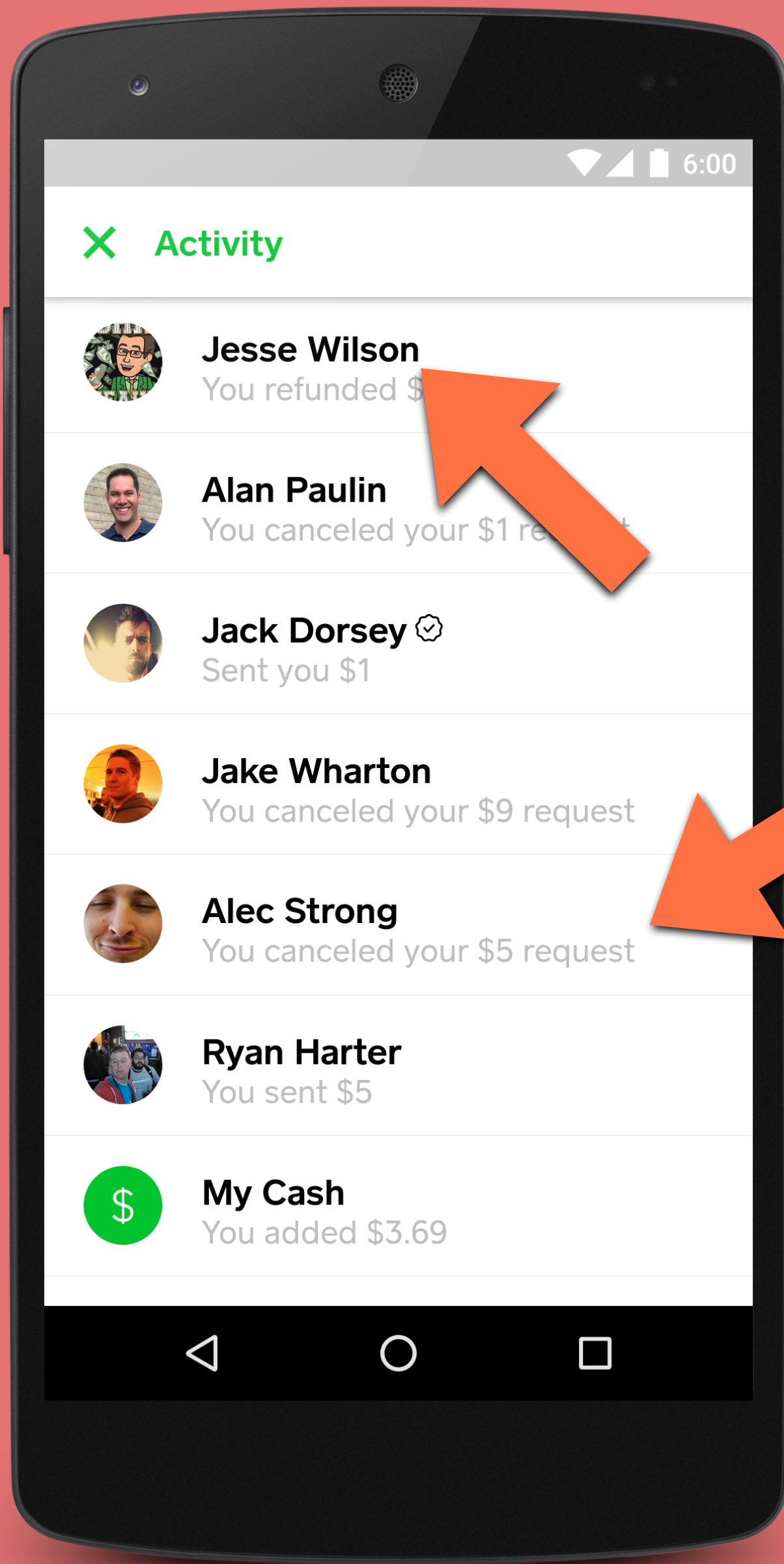


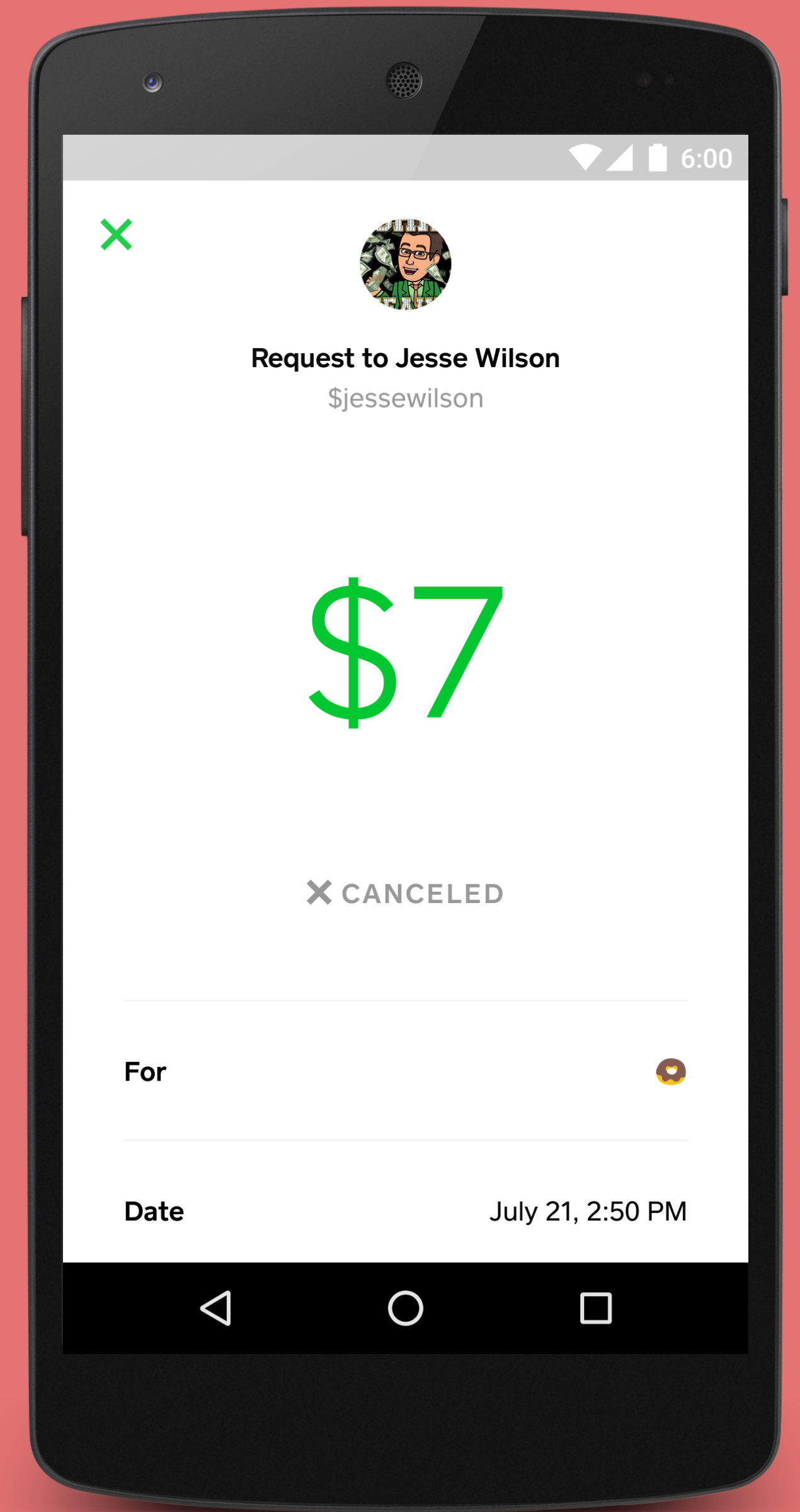
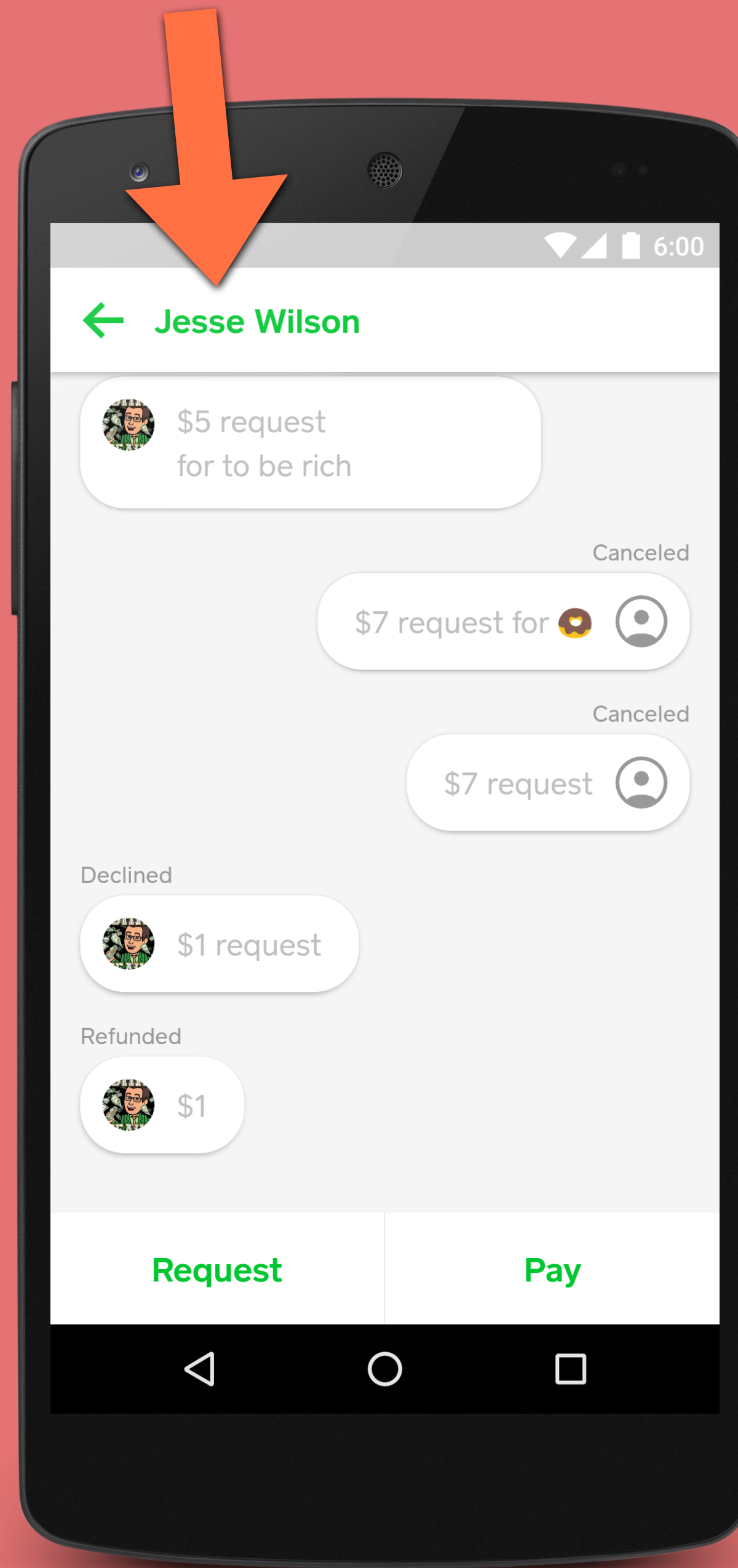
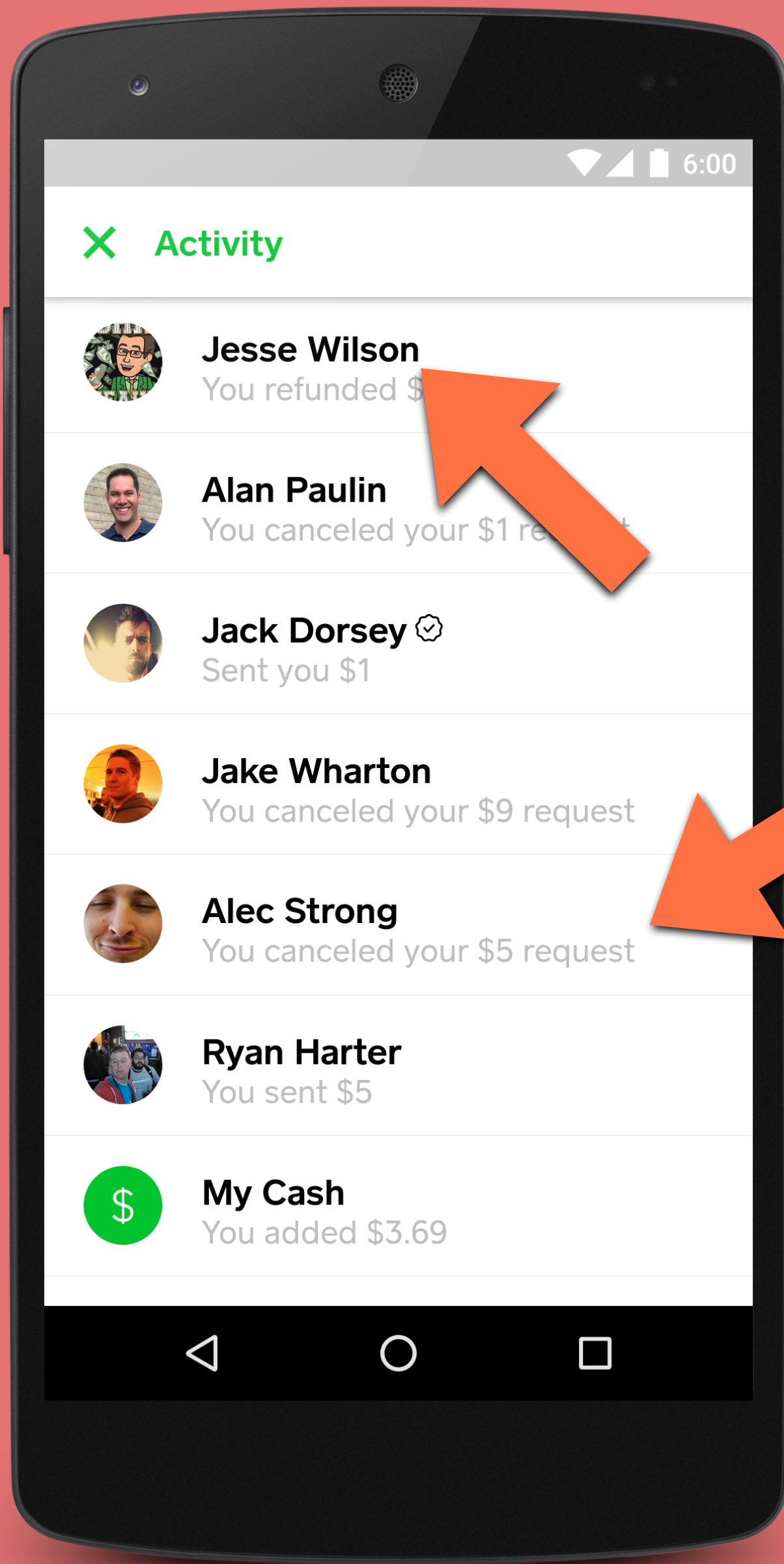
Square Cash: Activity v2

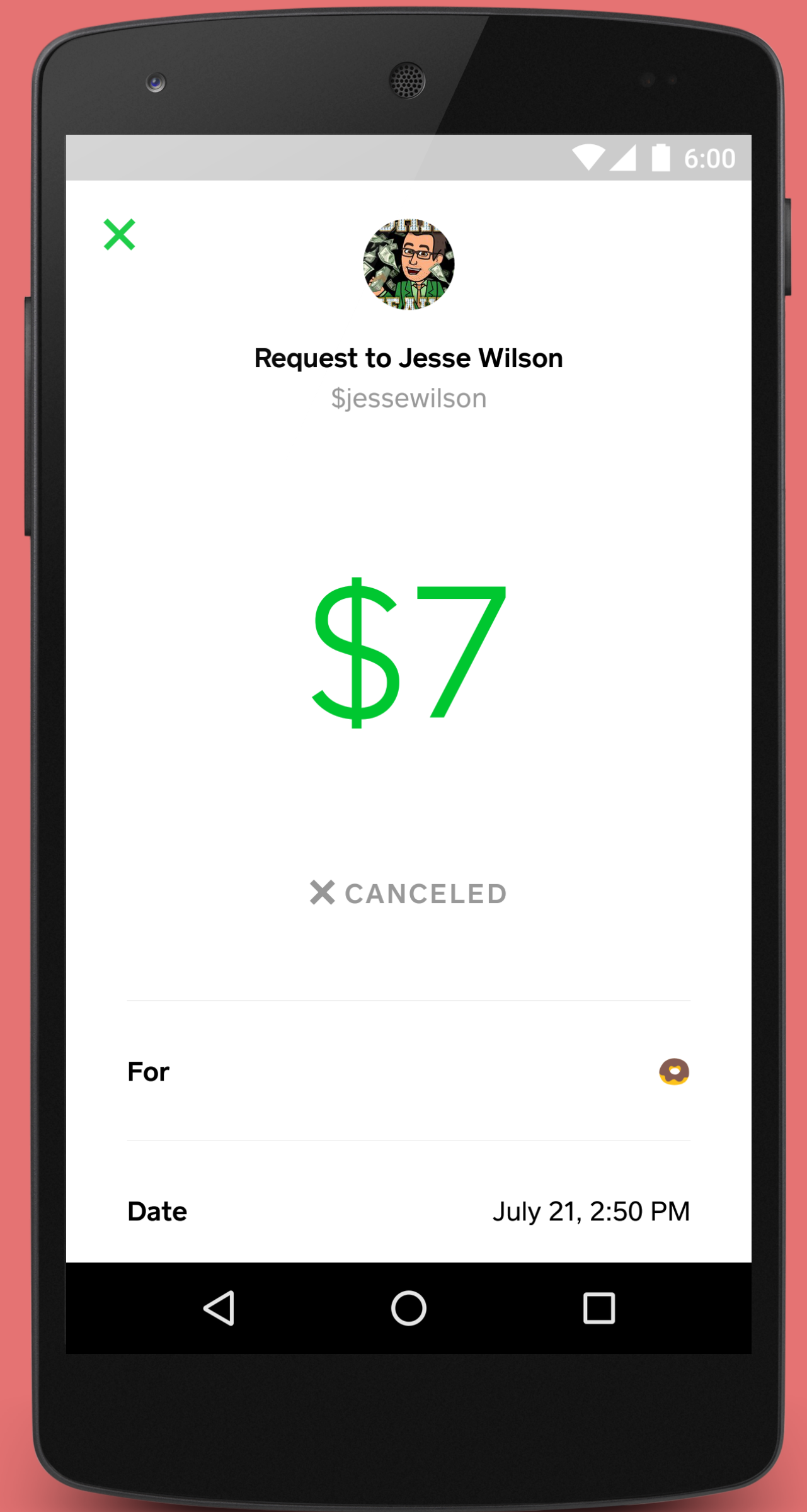
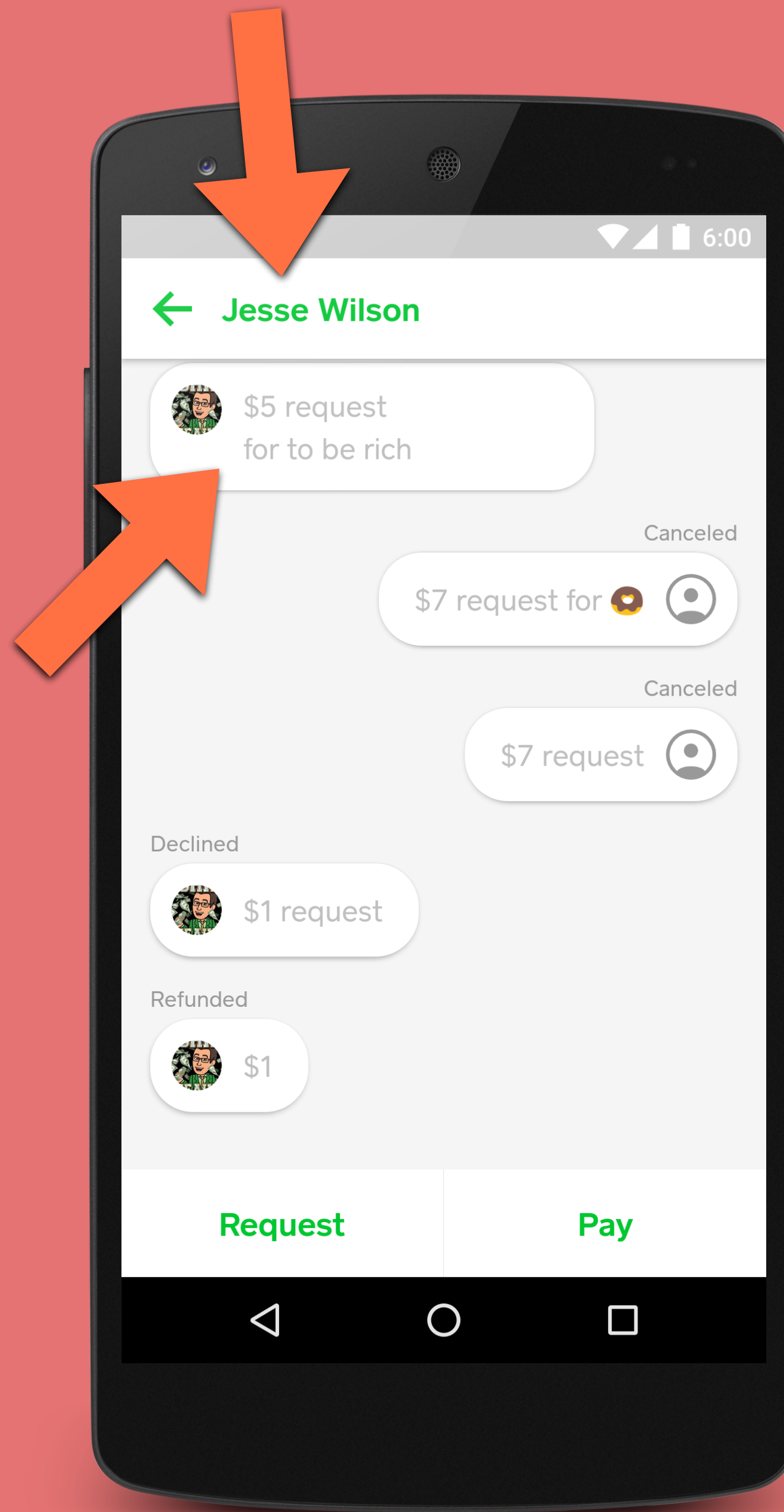
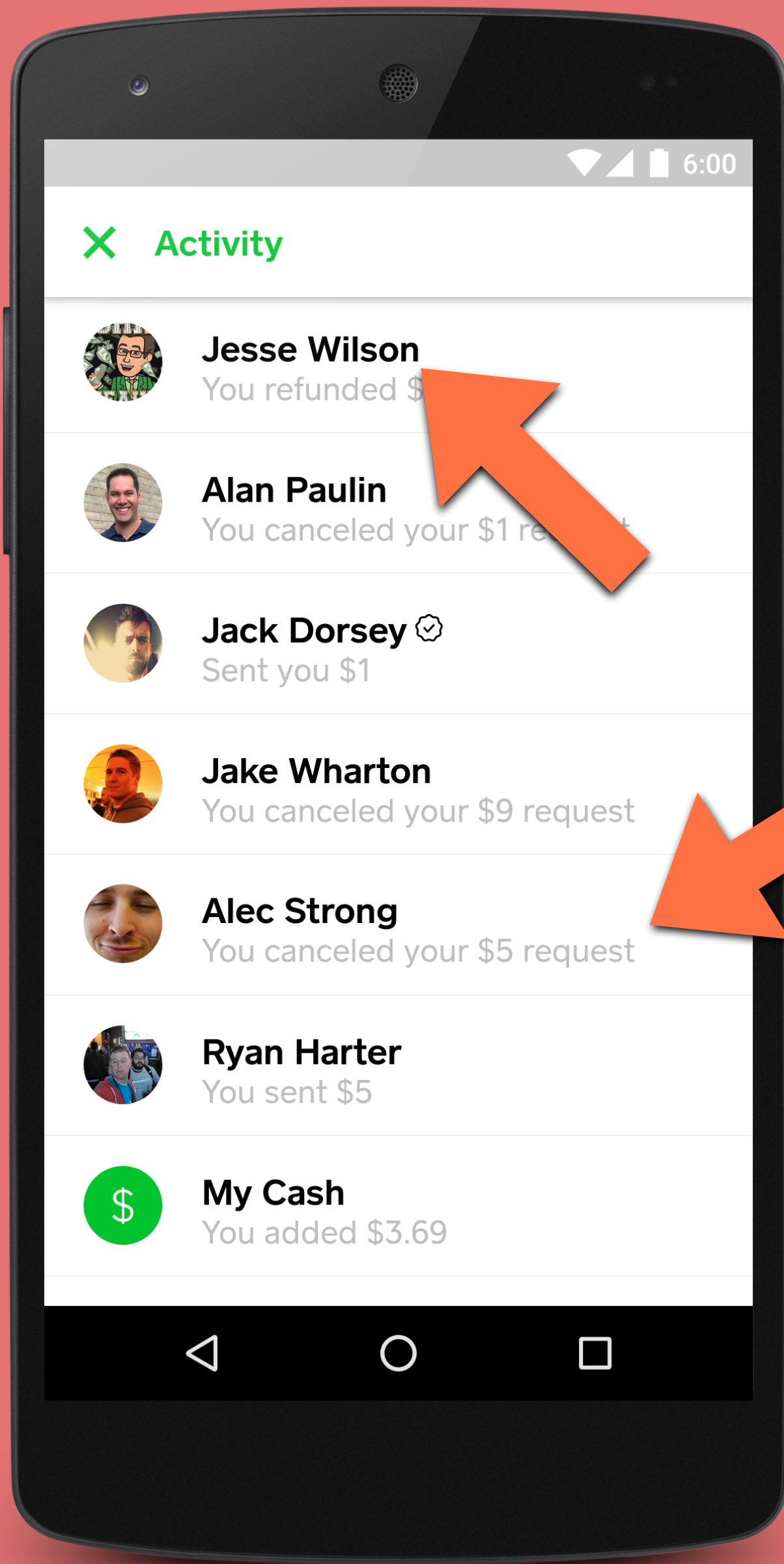


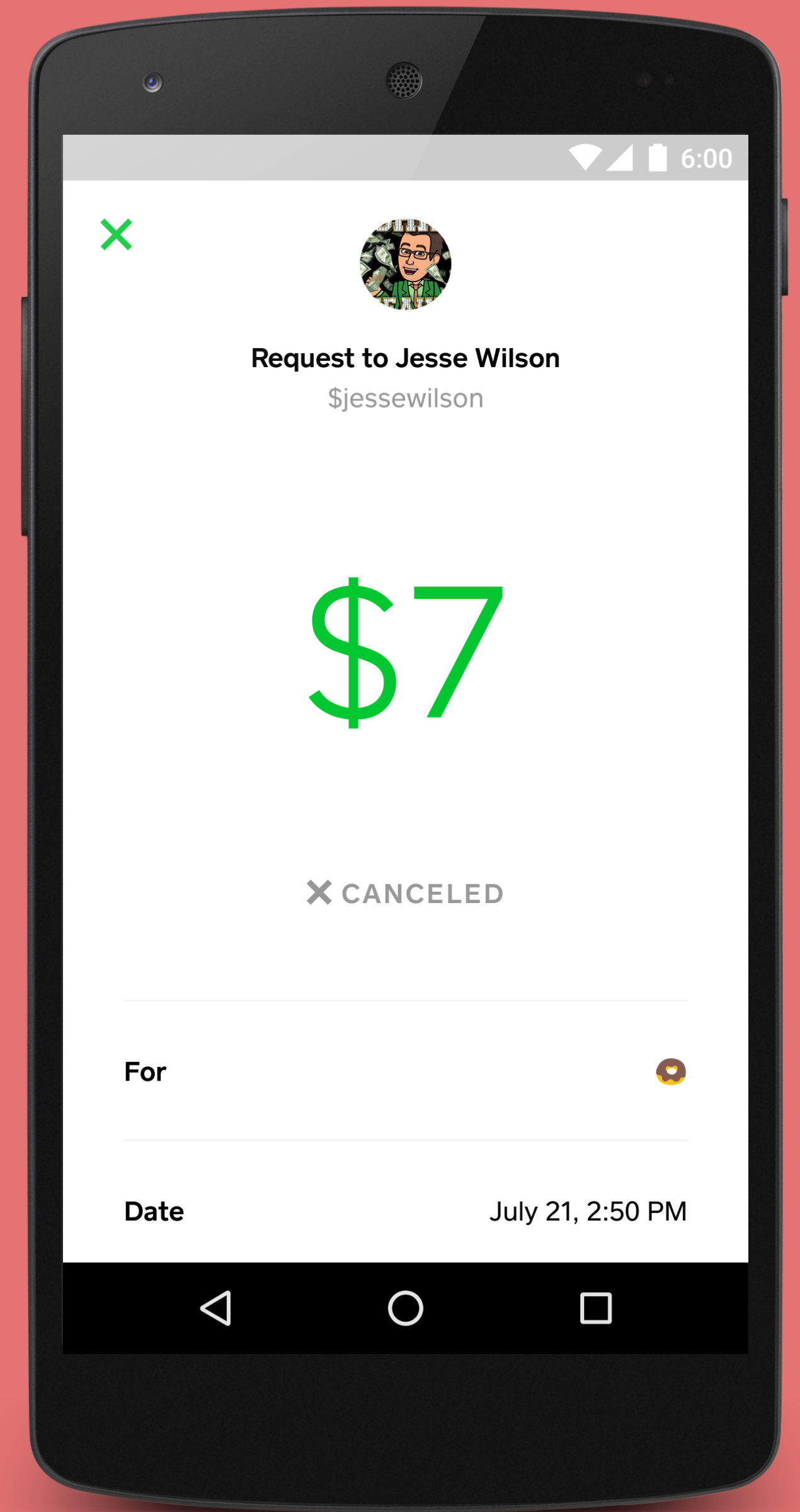
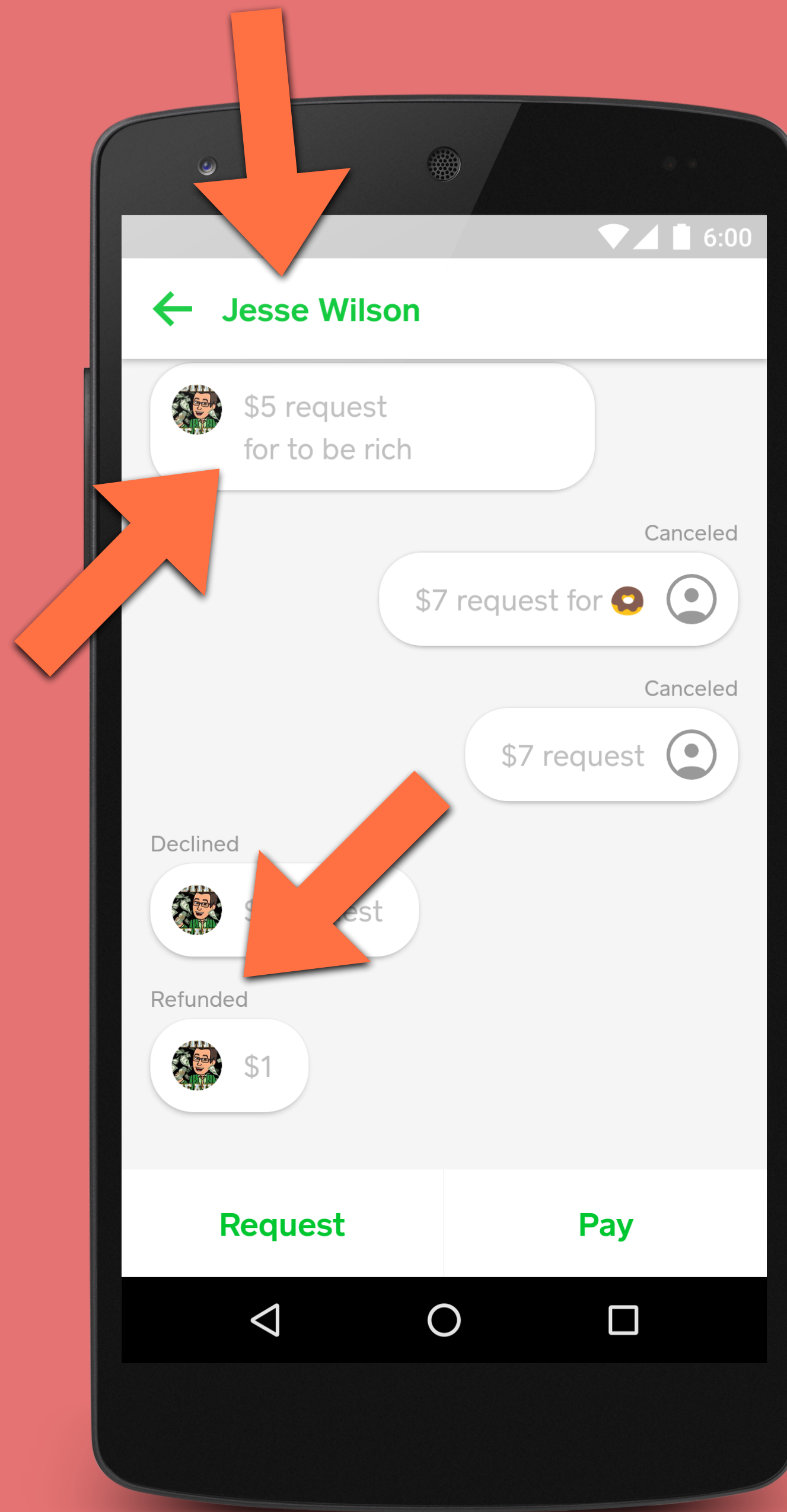
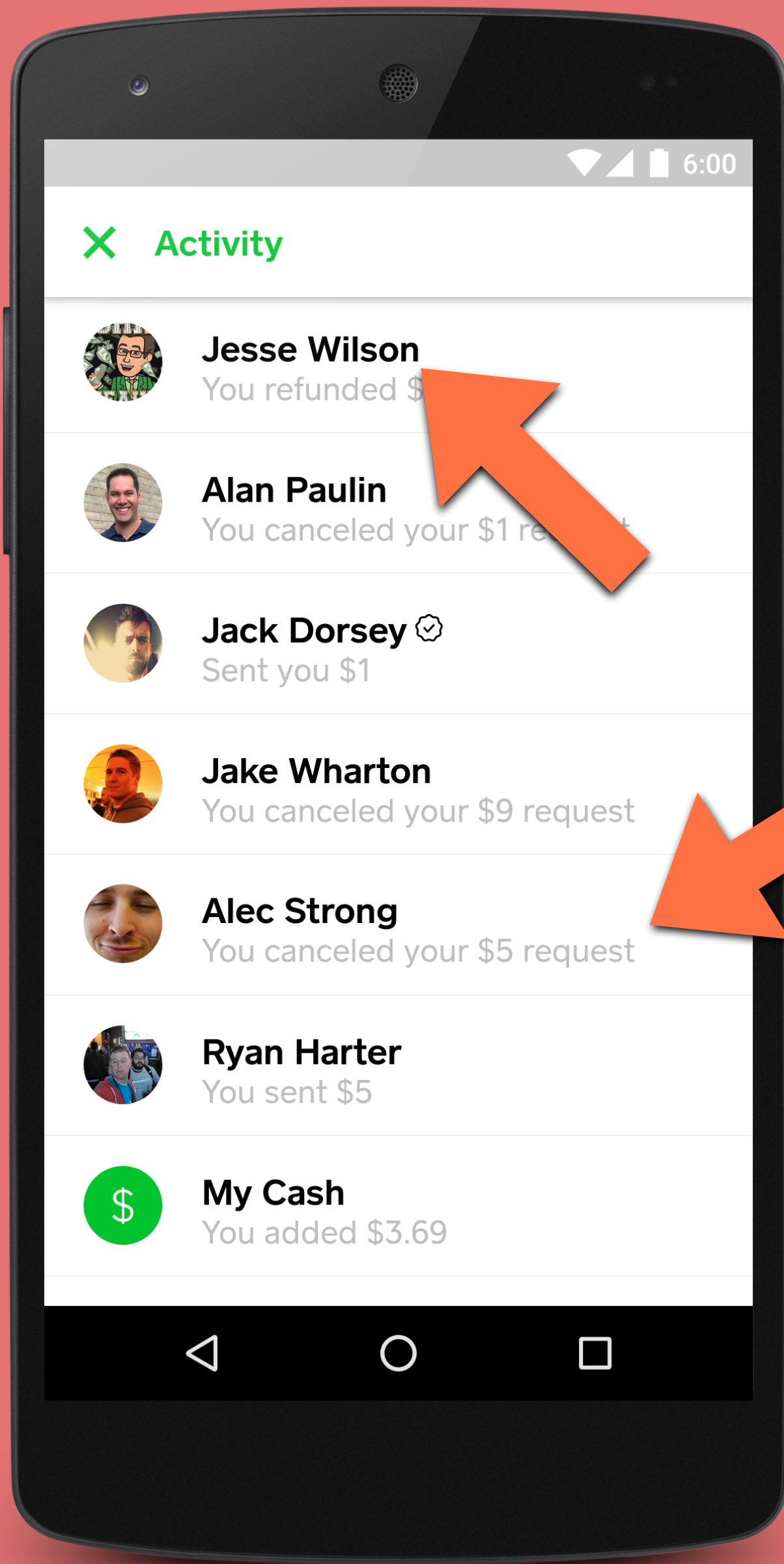


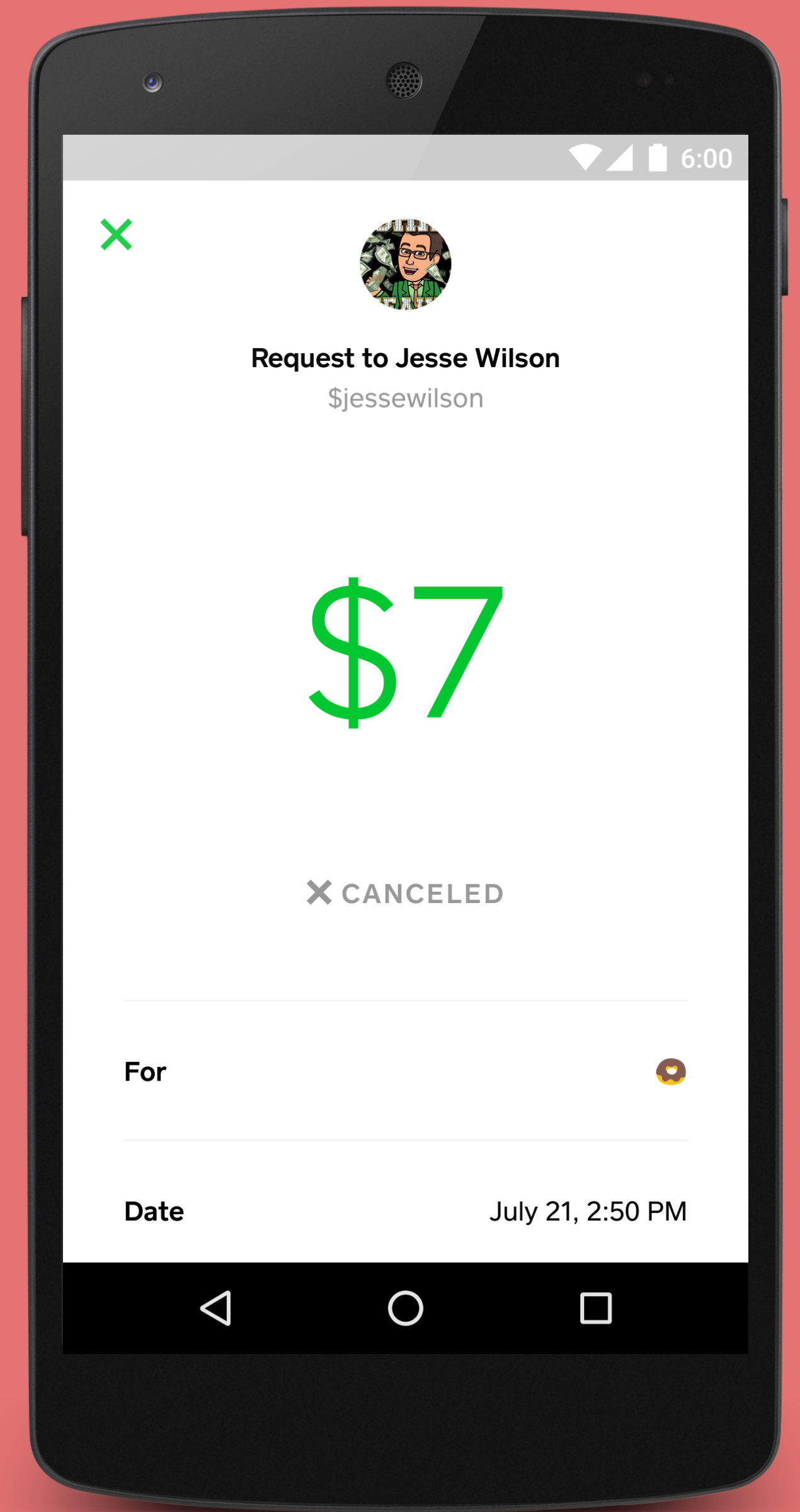
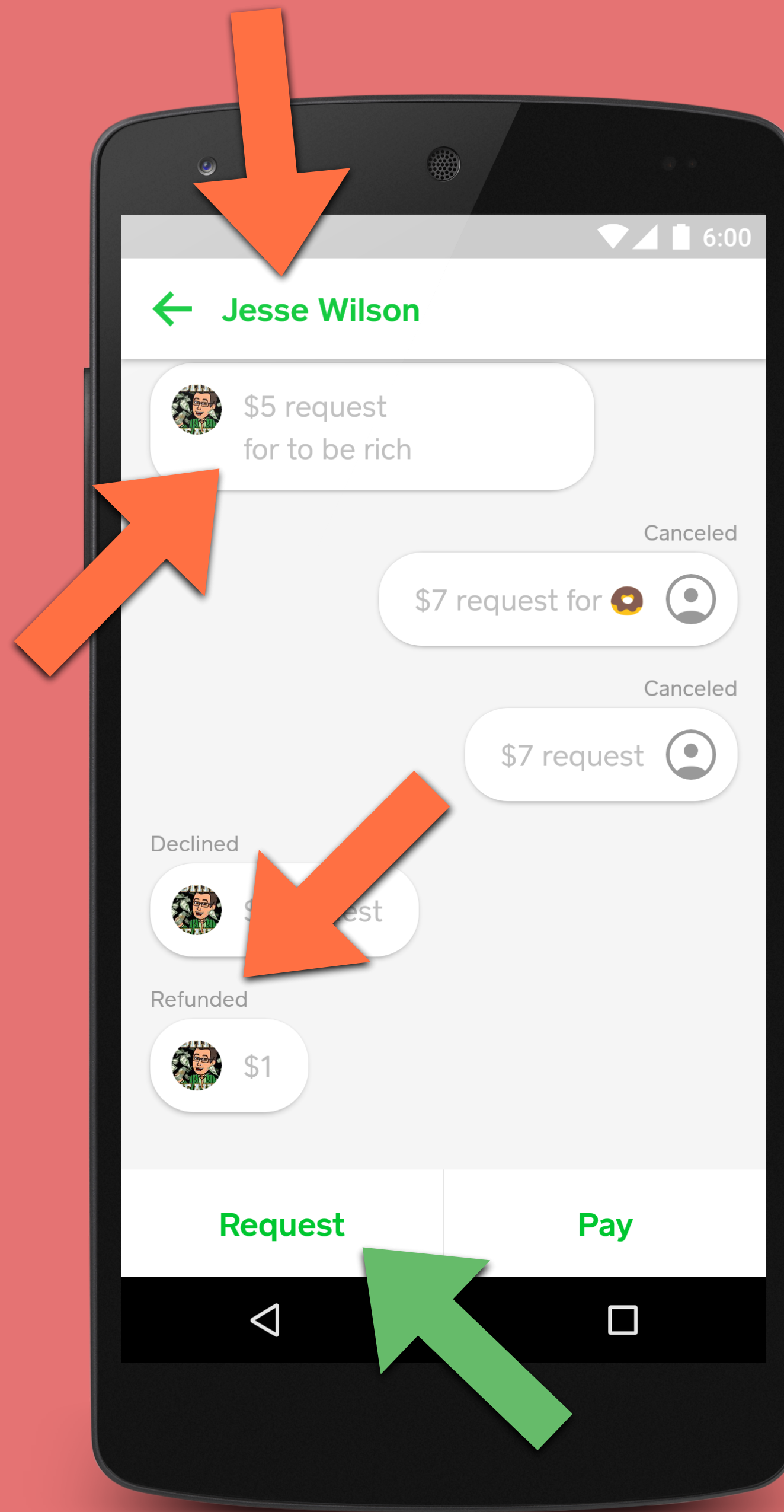
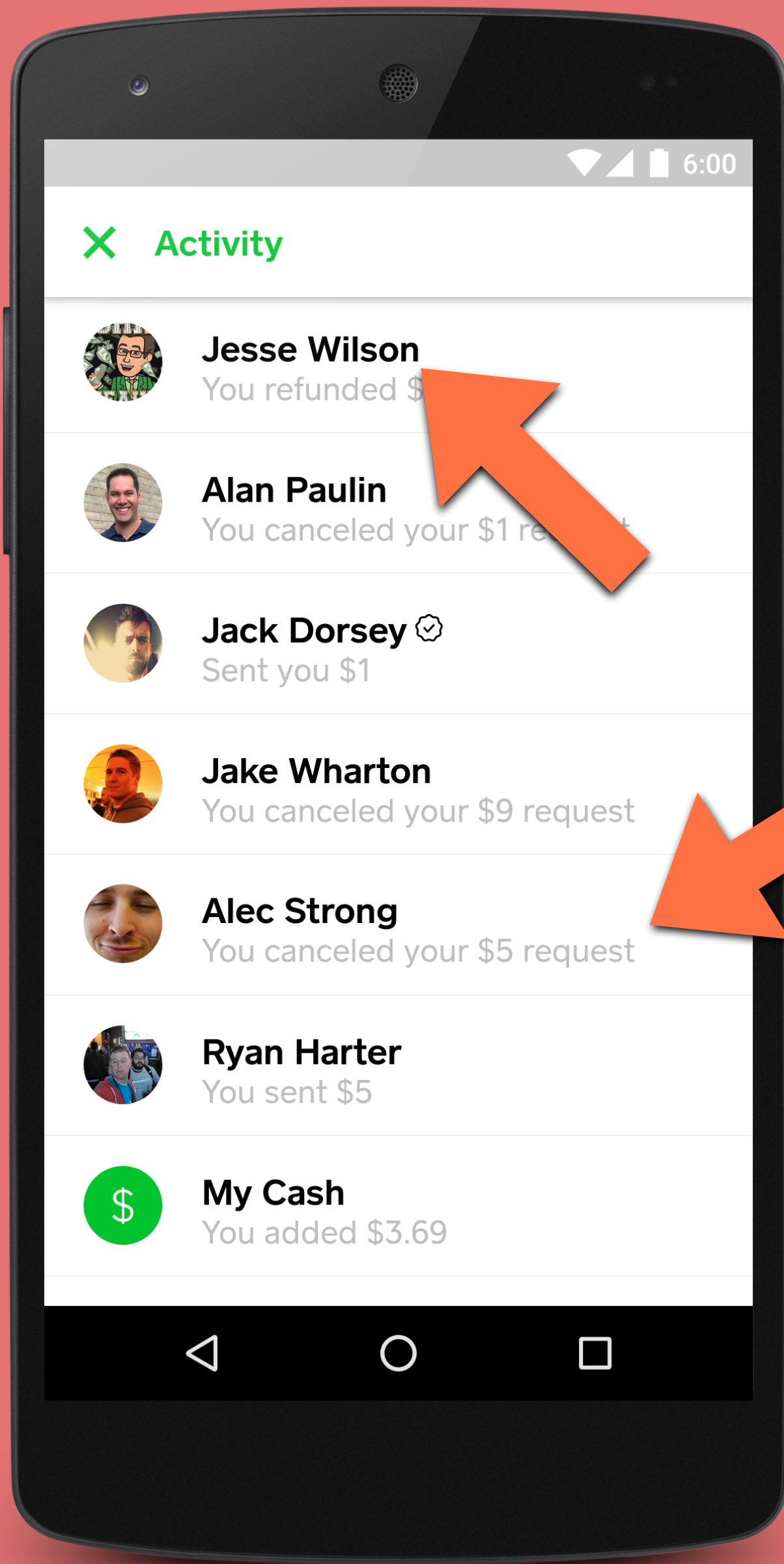


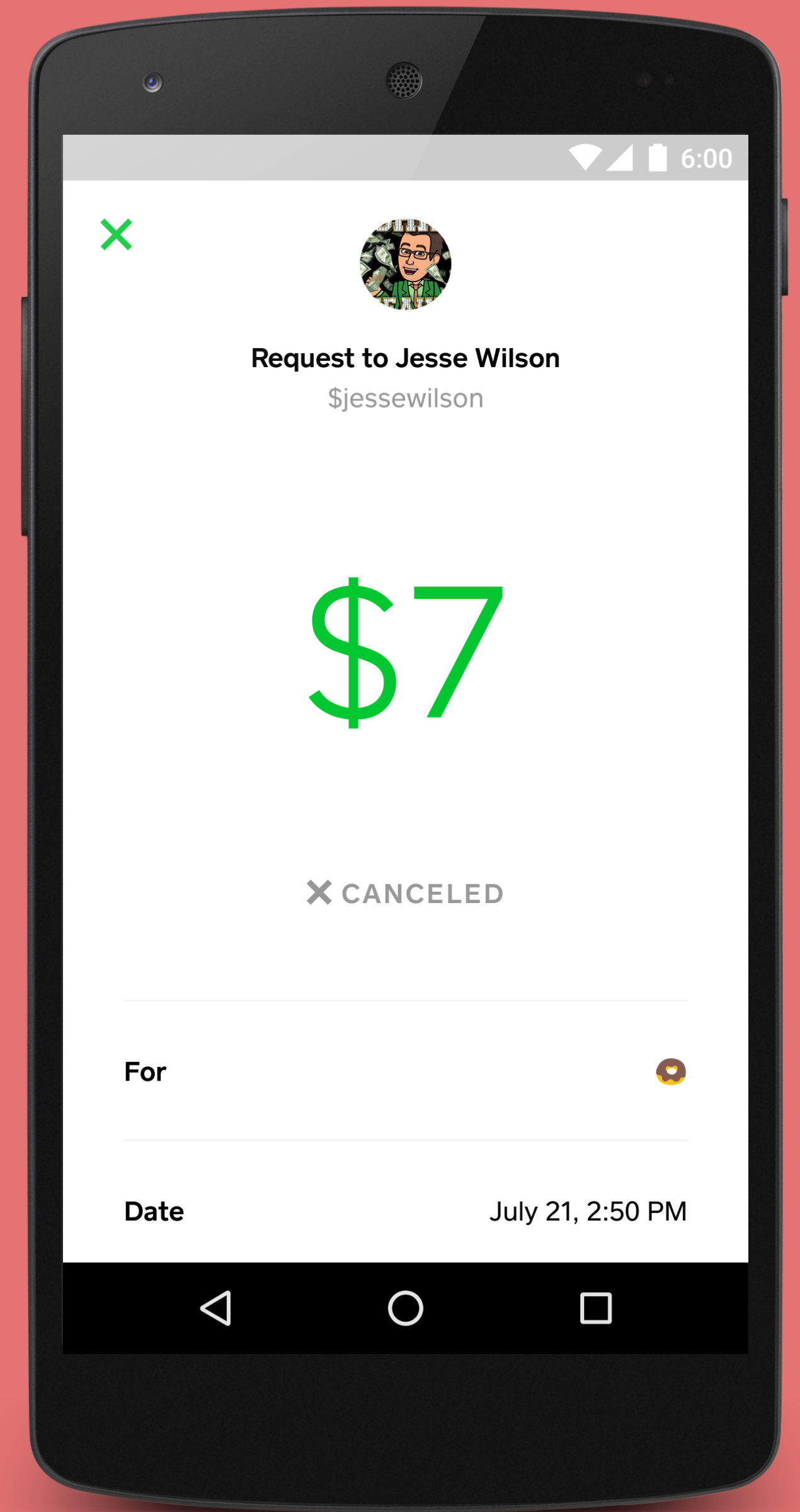
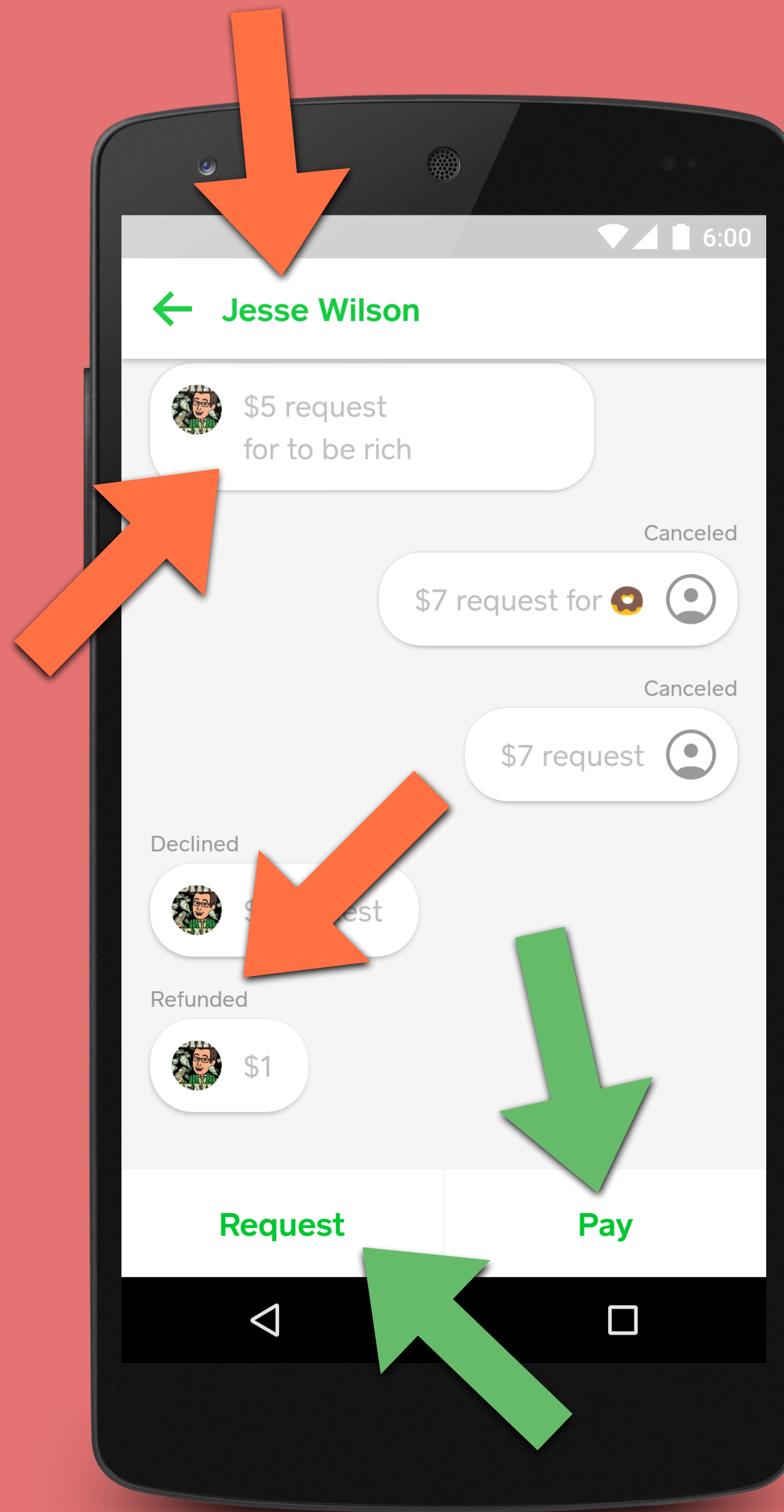
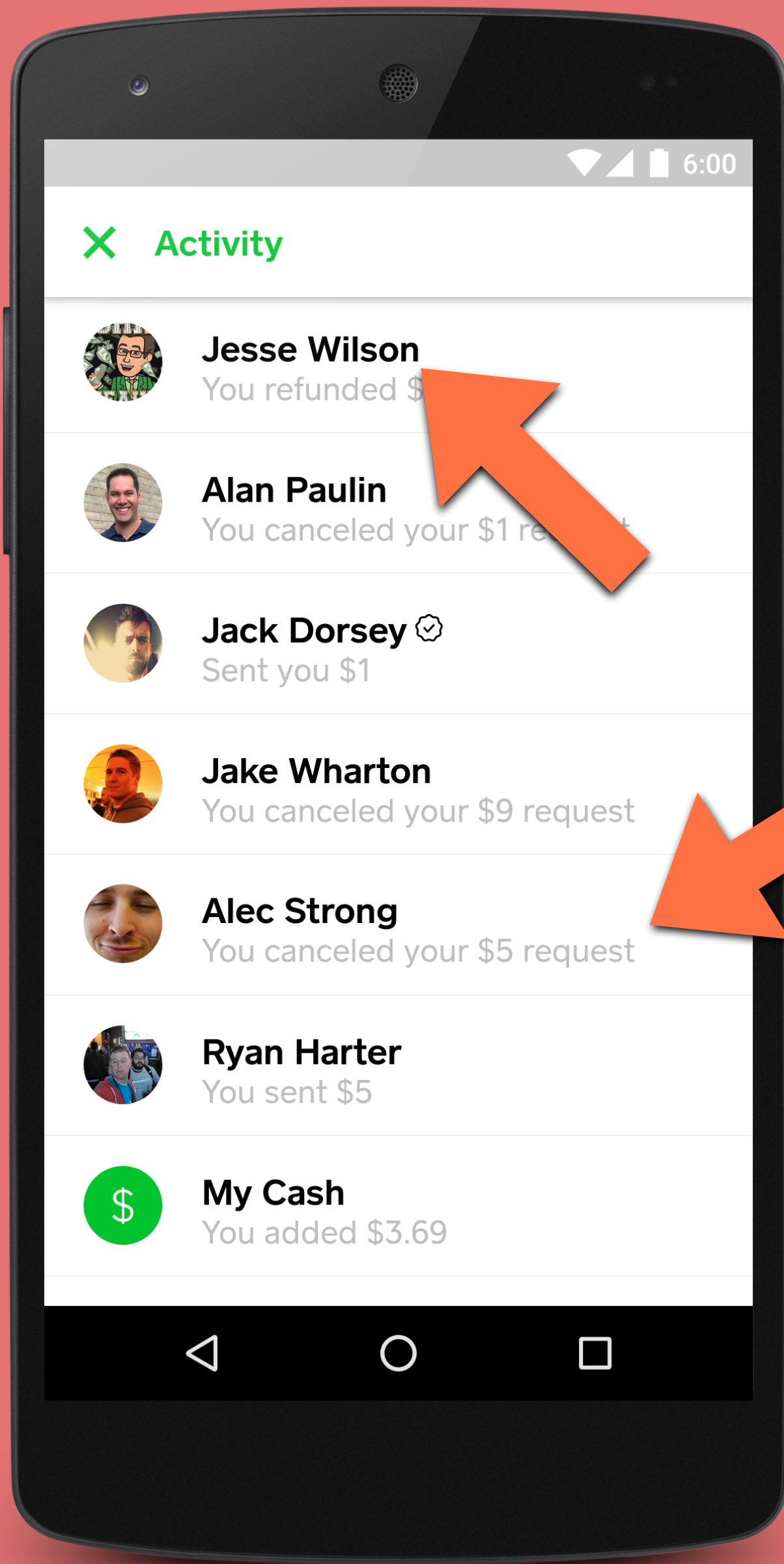


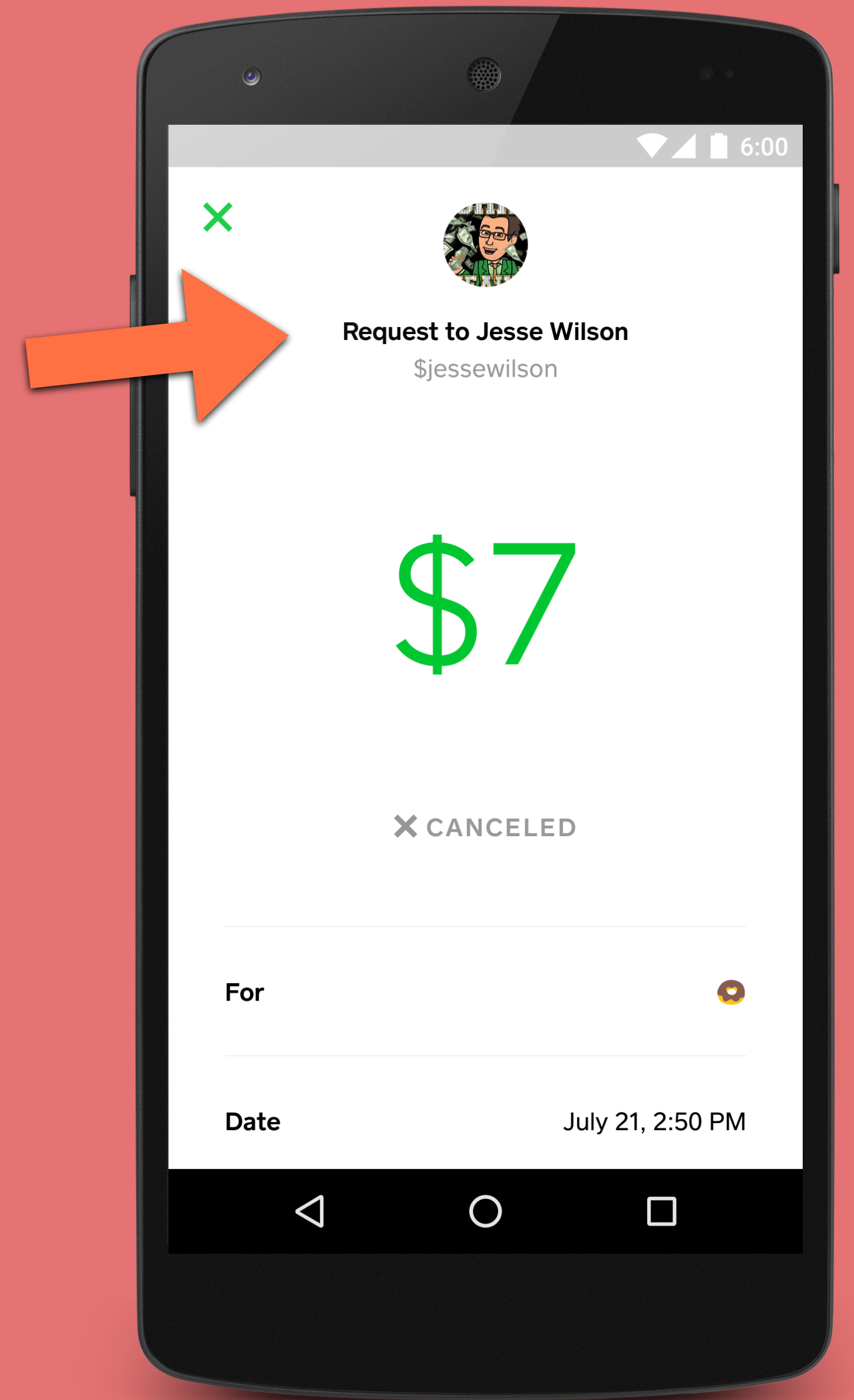
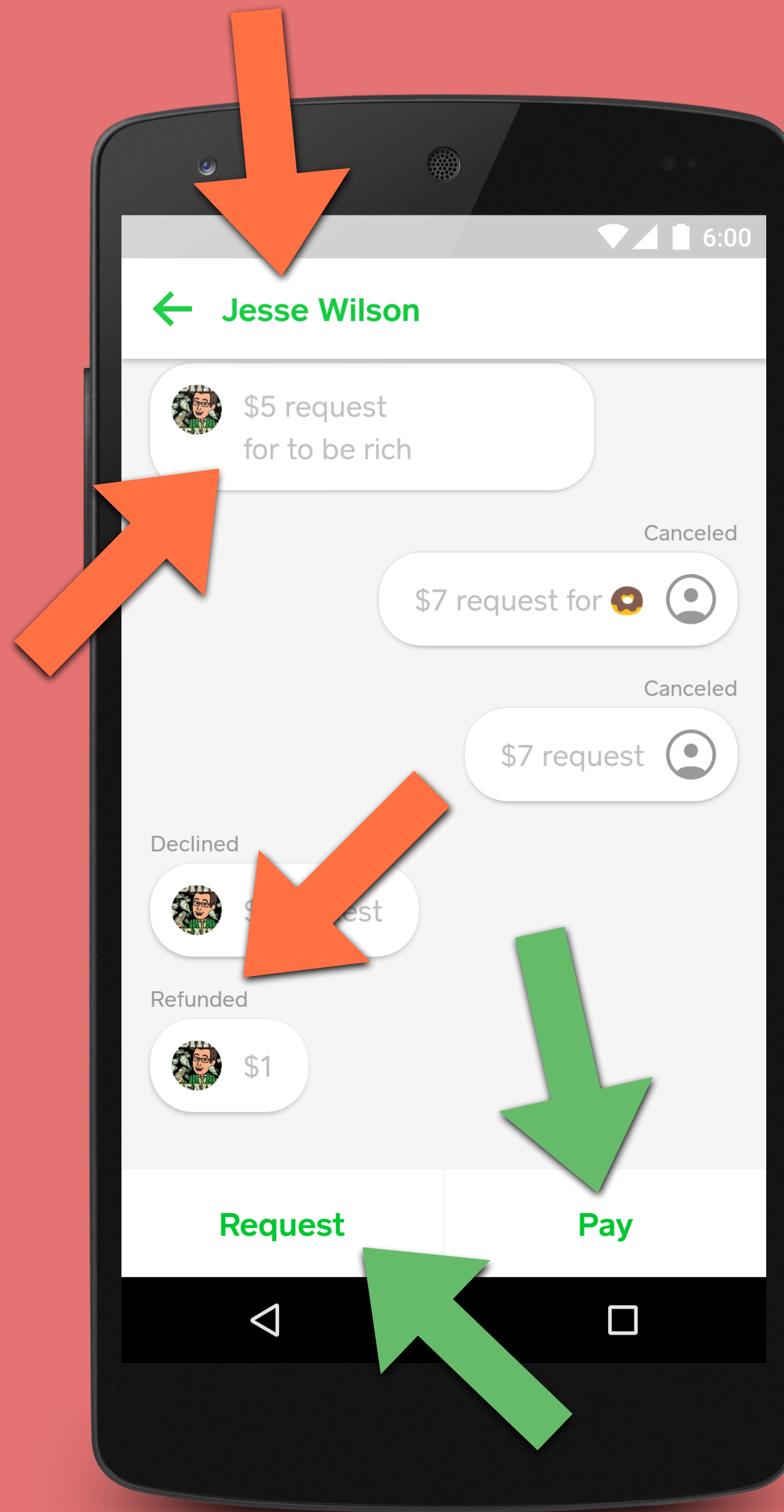
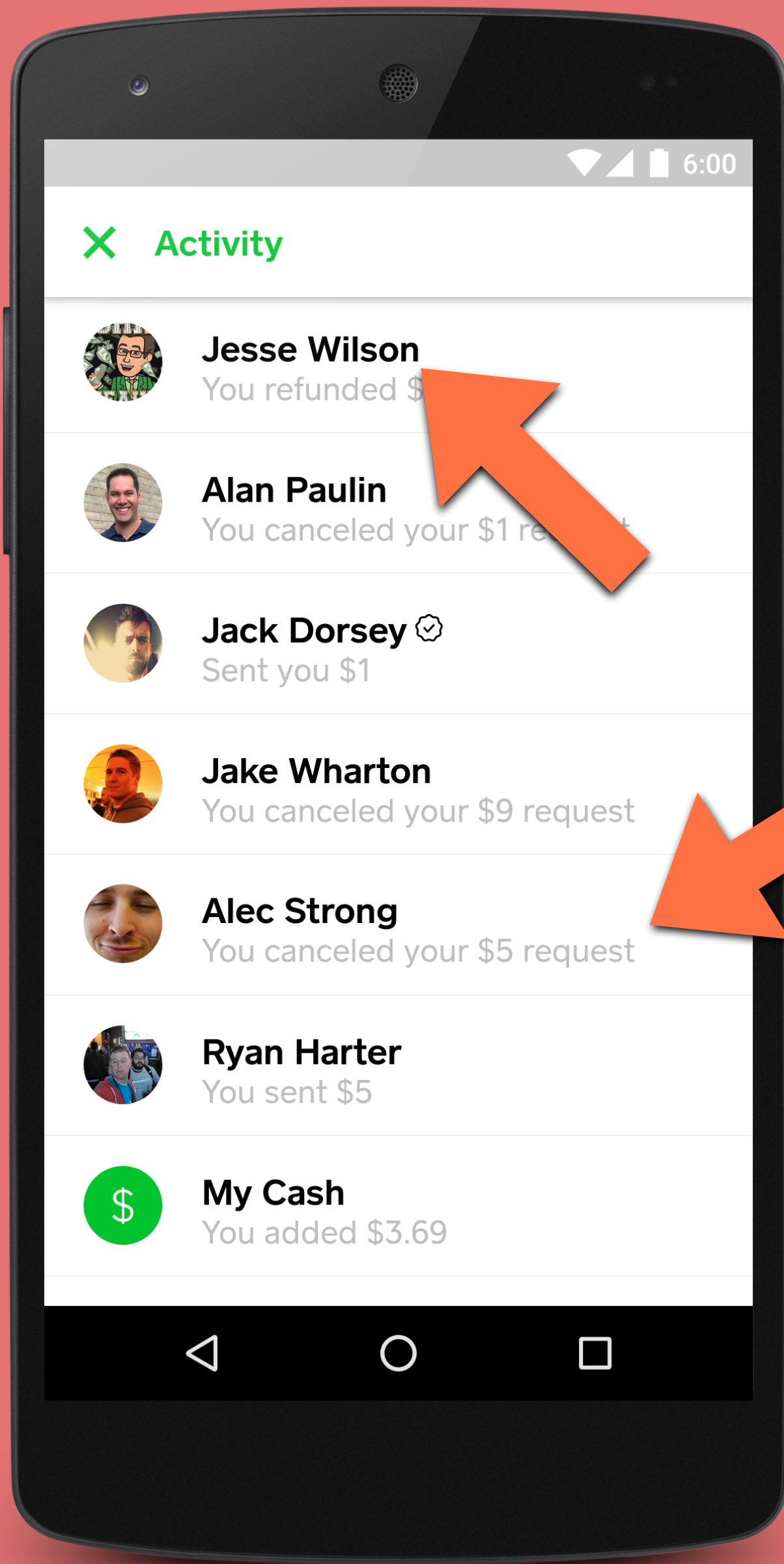


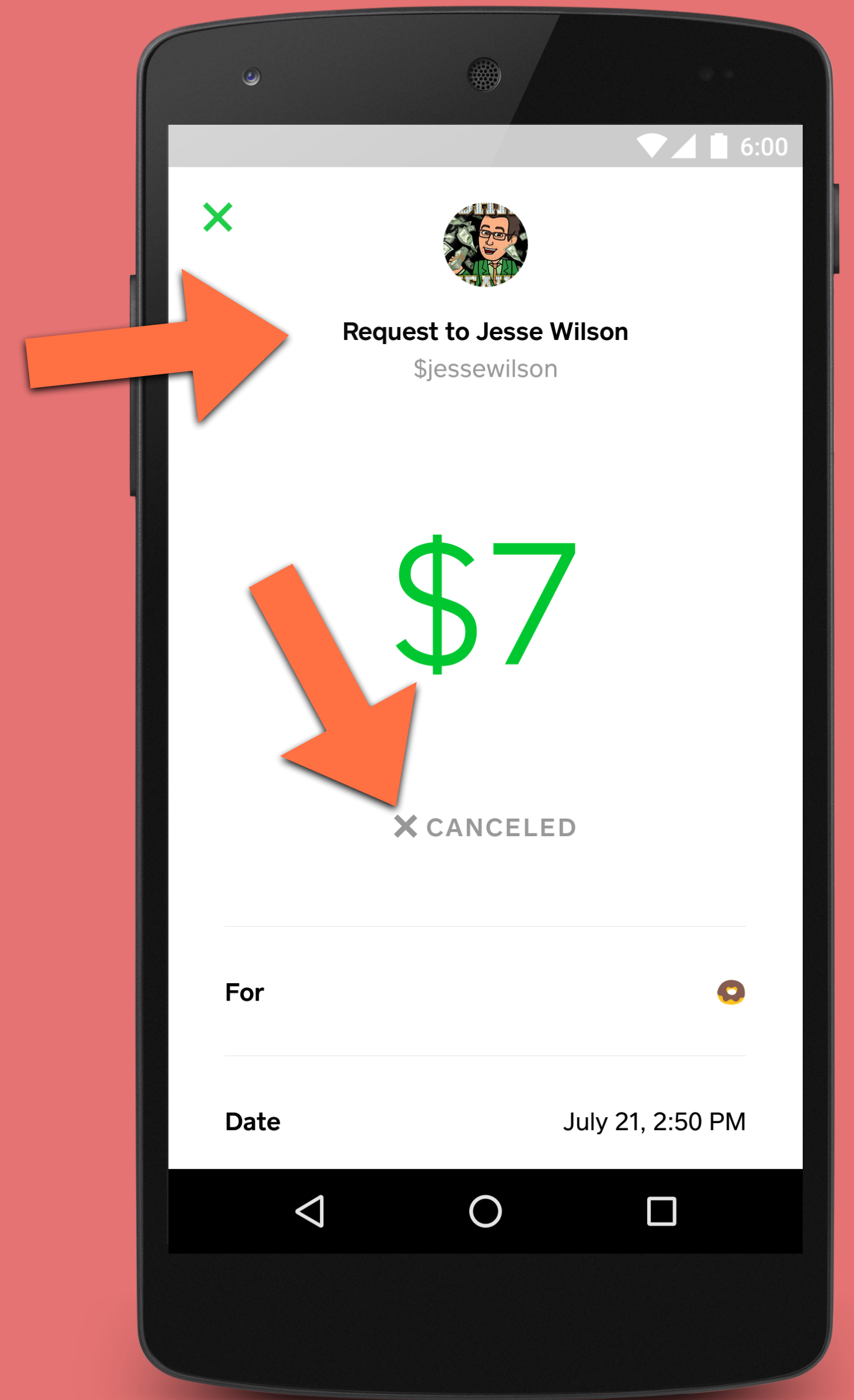
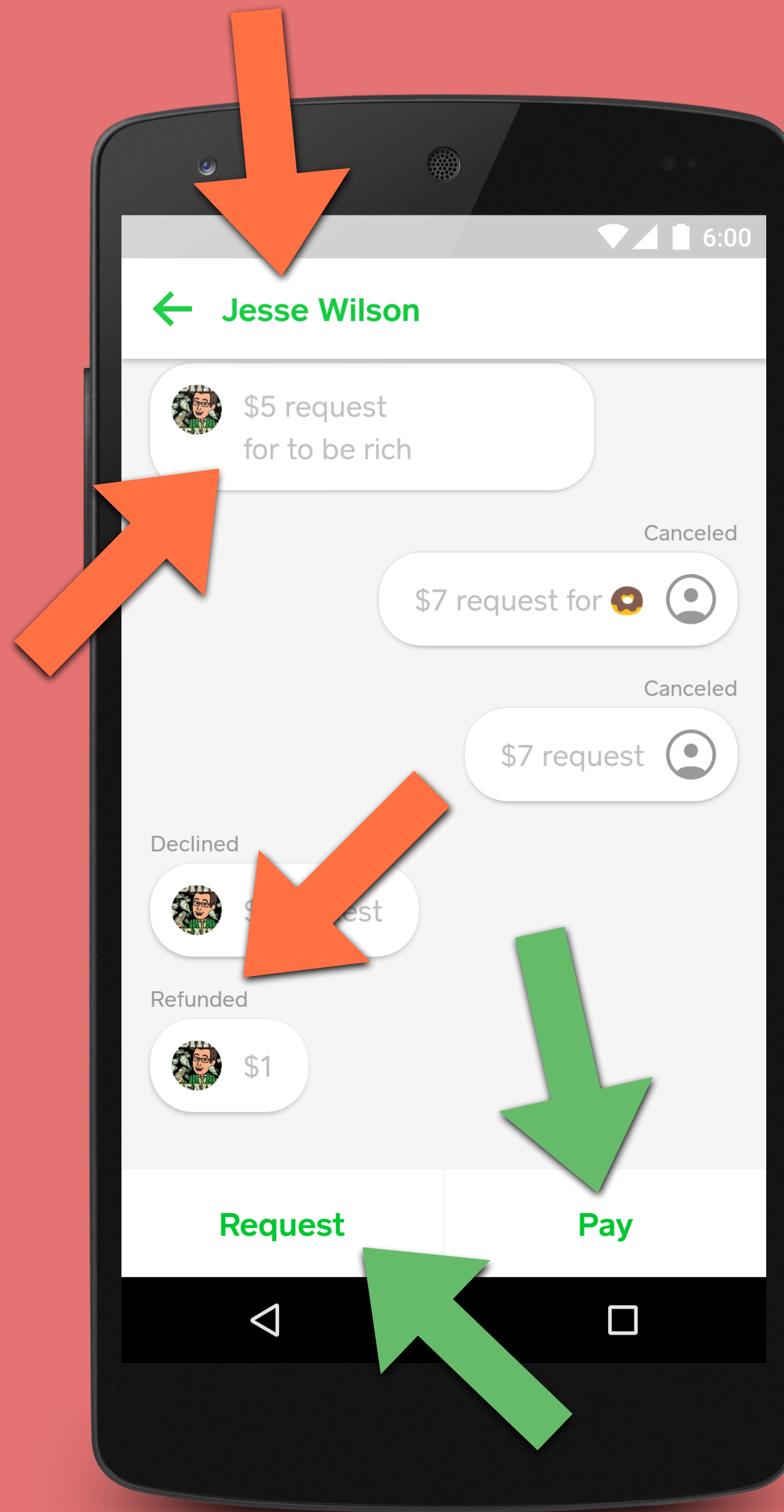
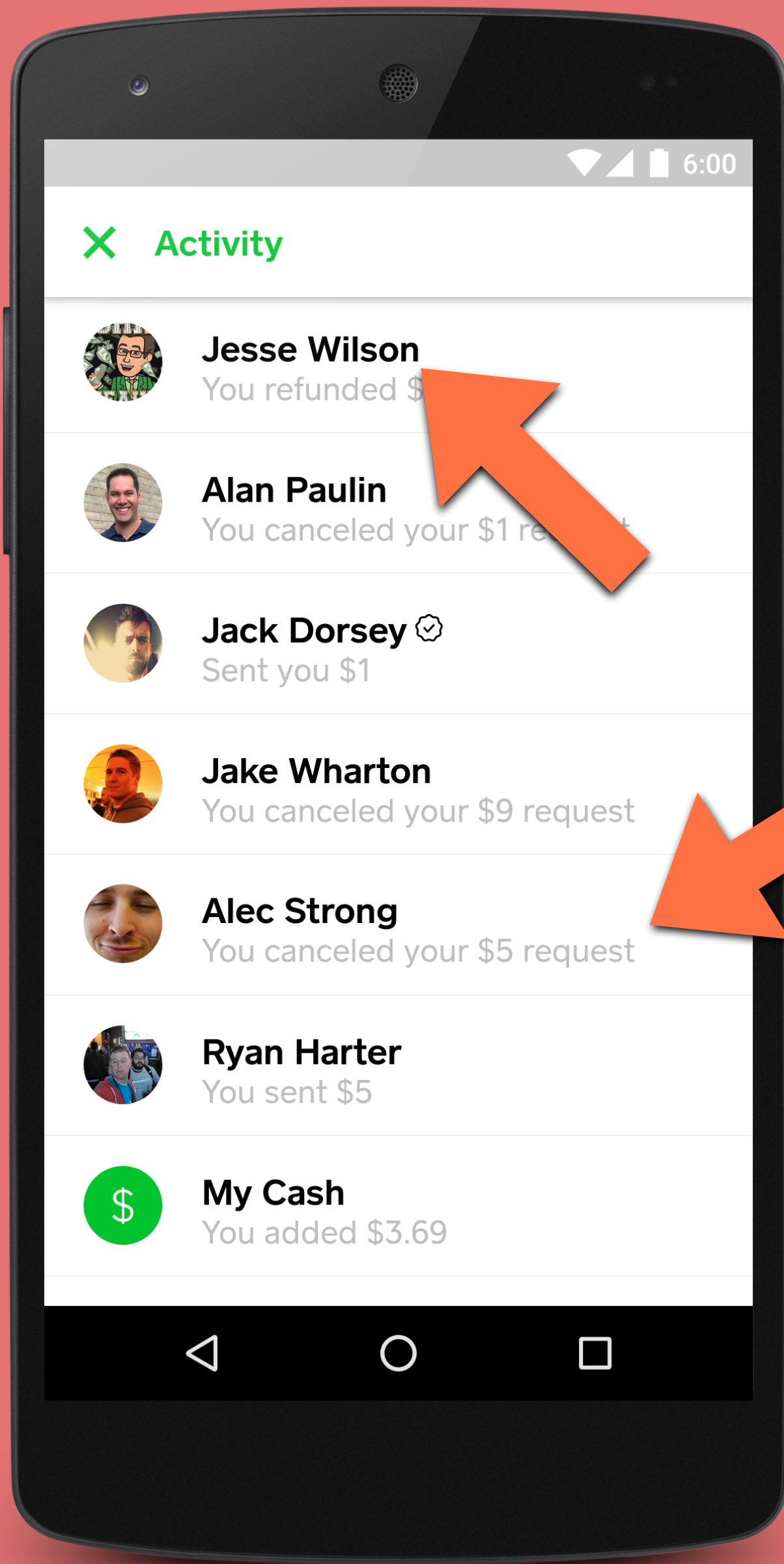


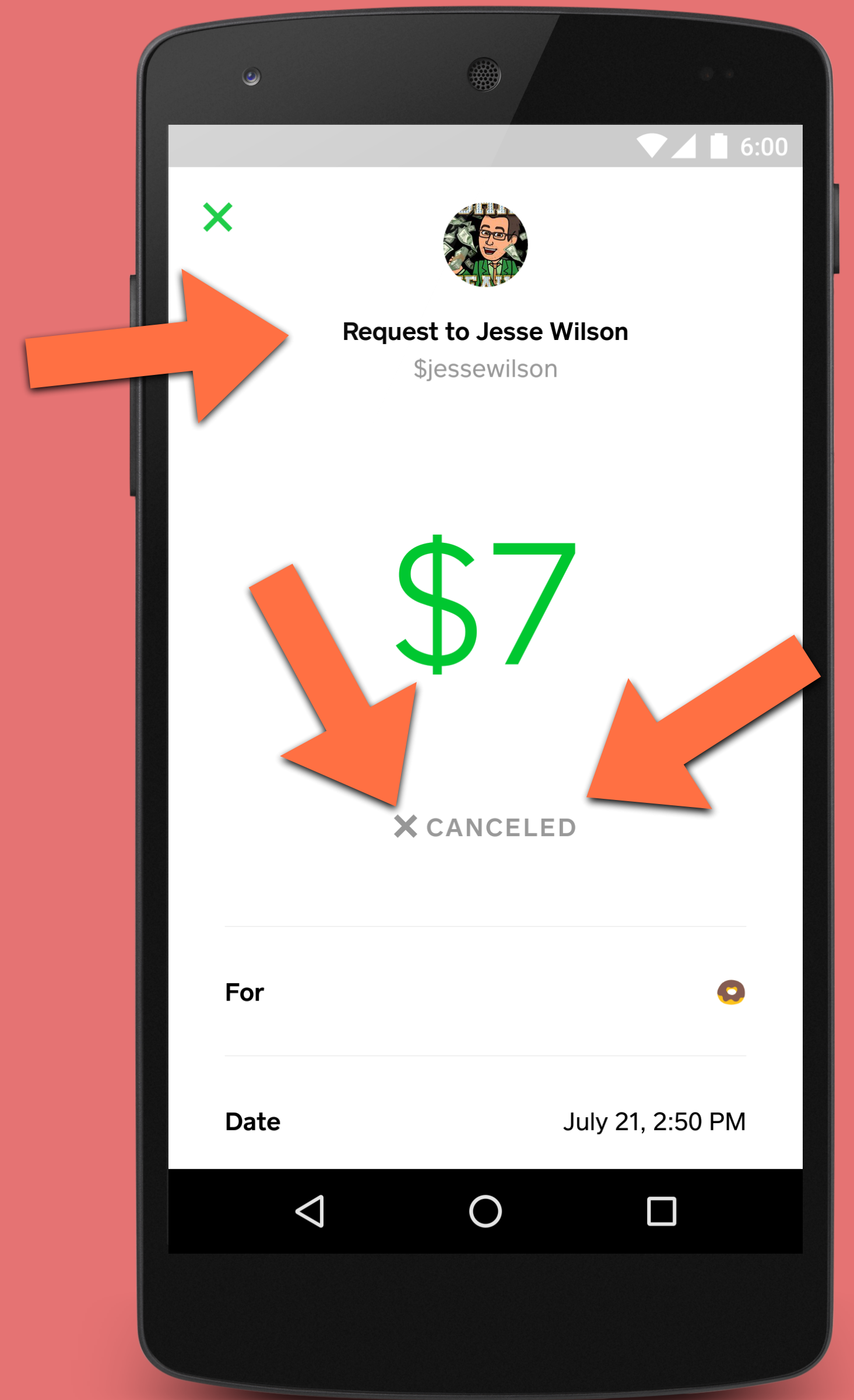
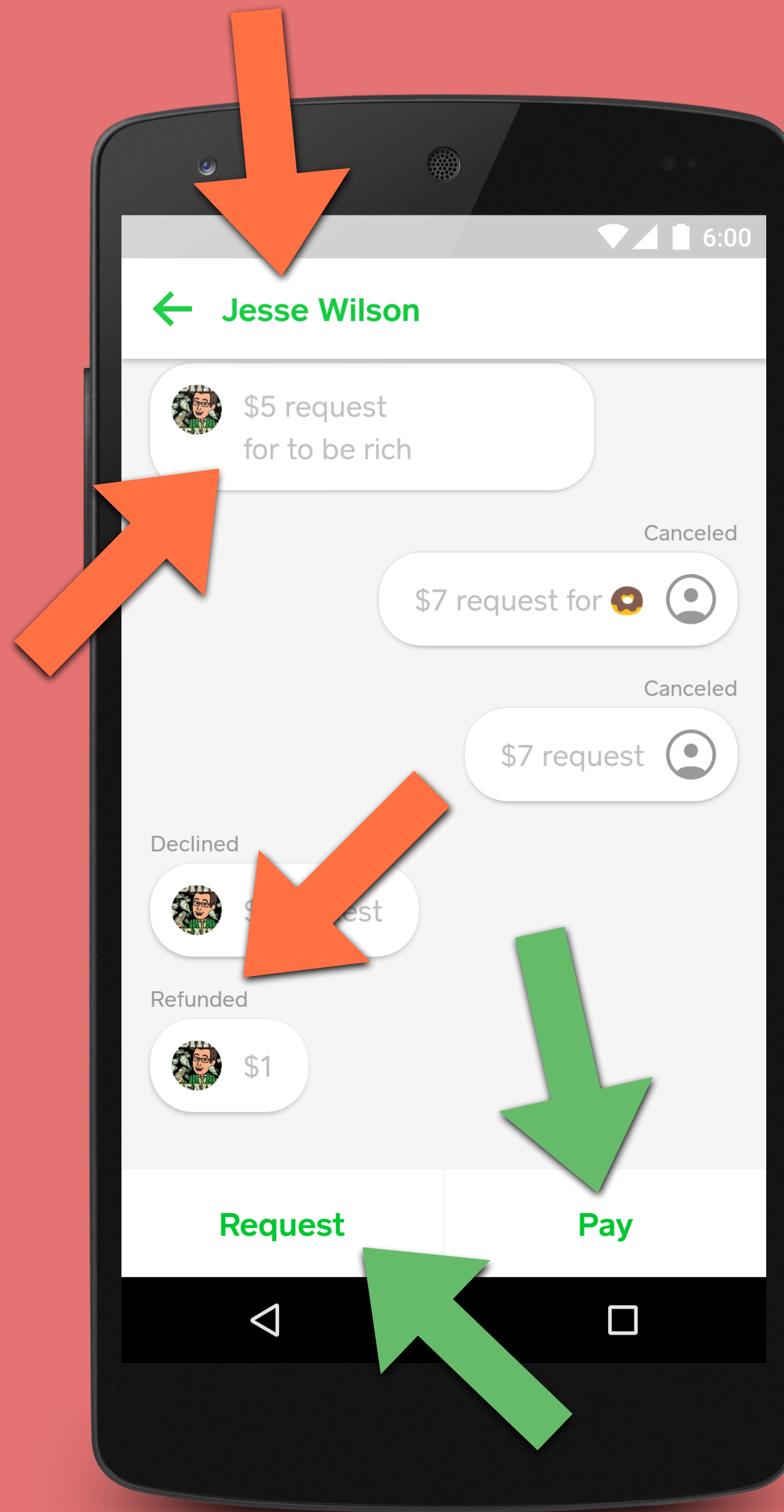
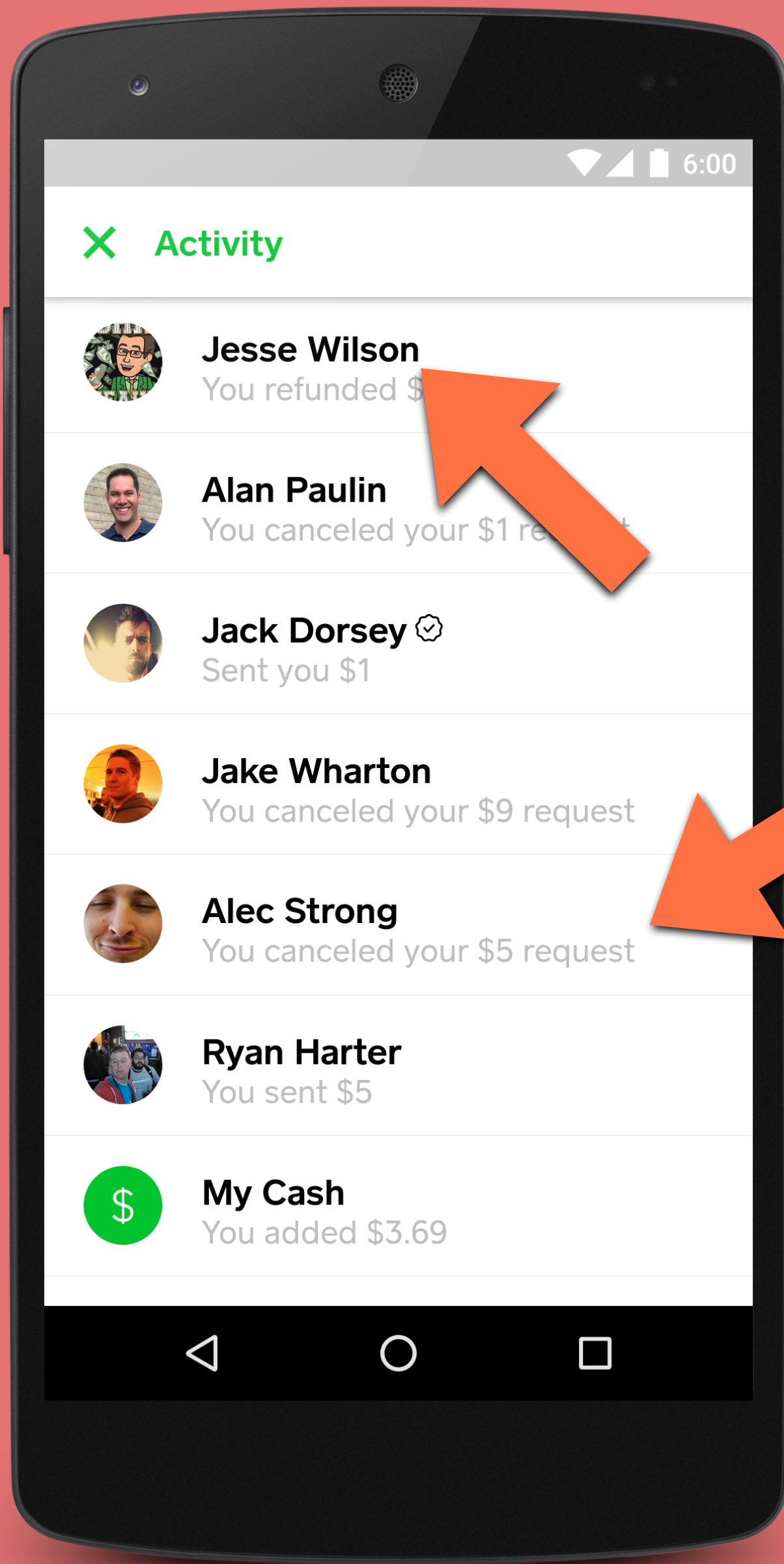


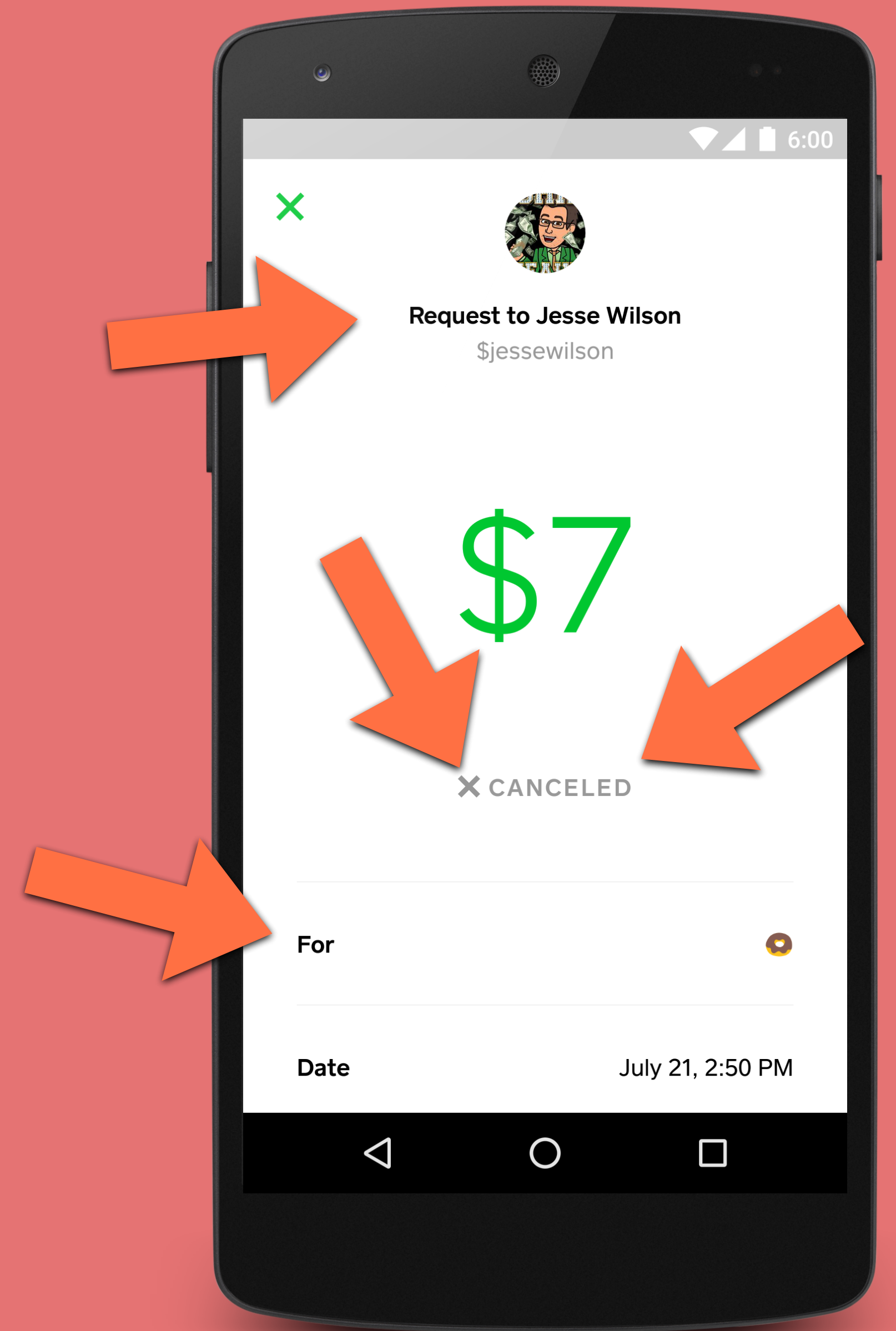
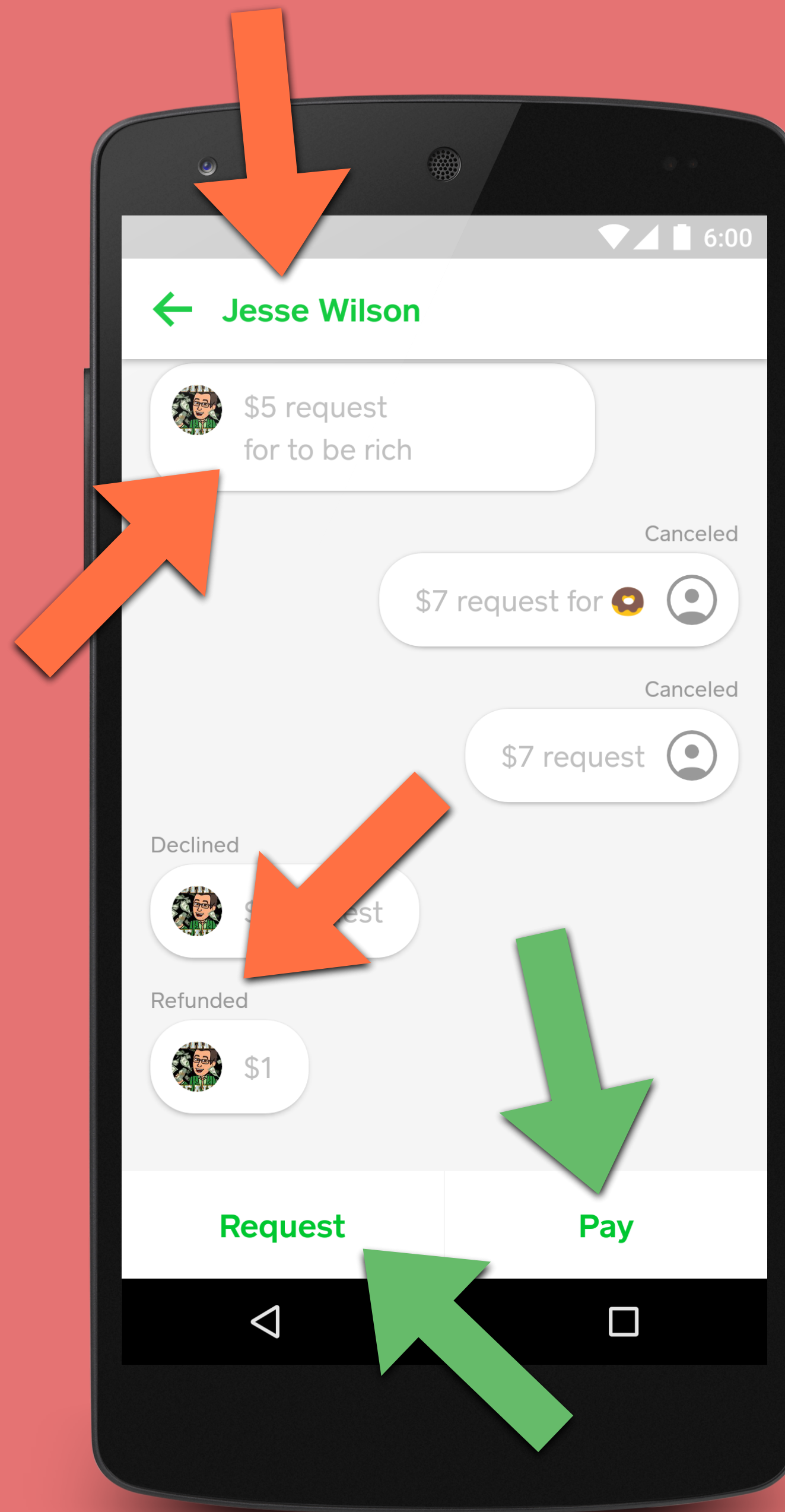
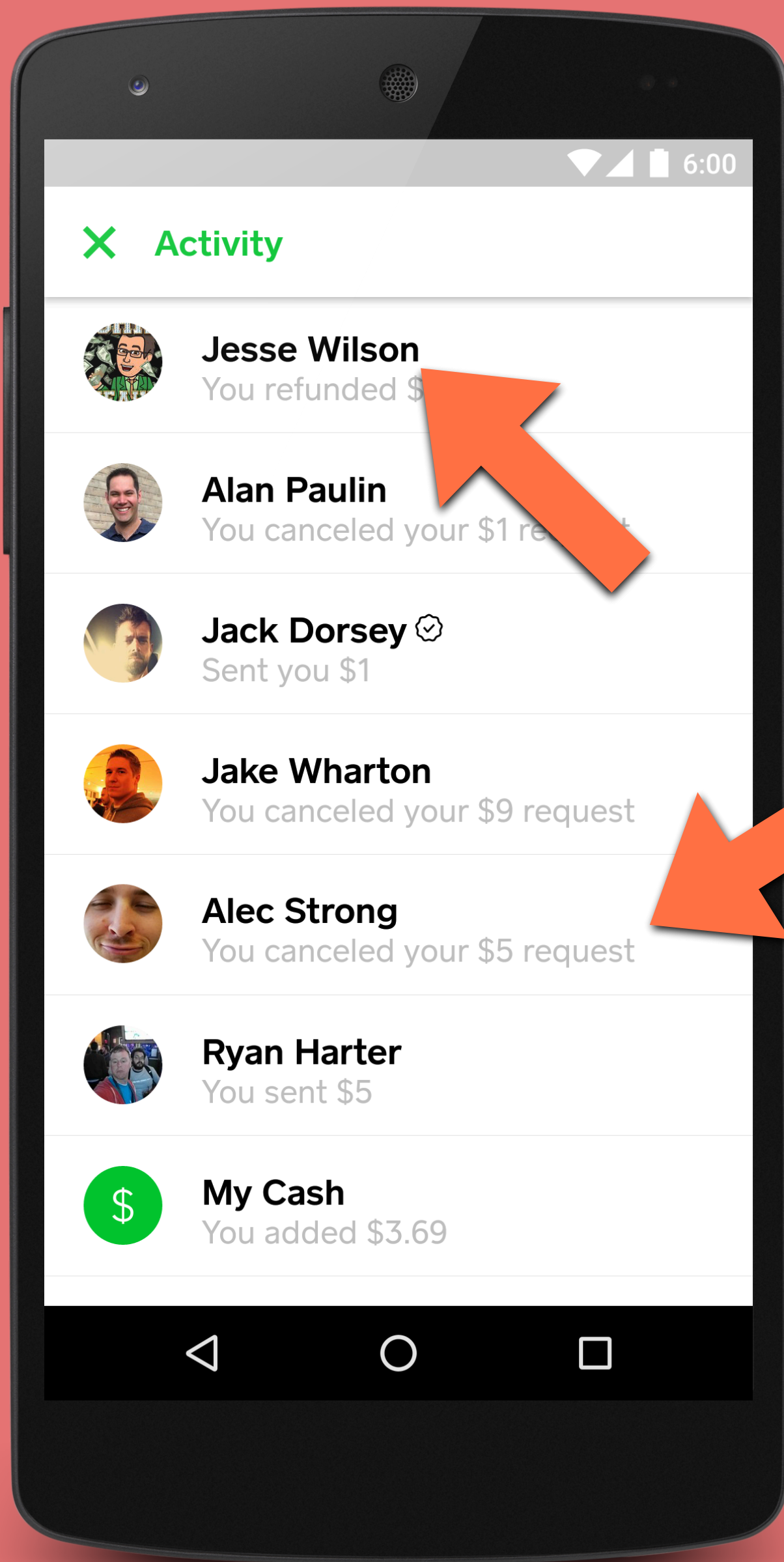


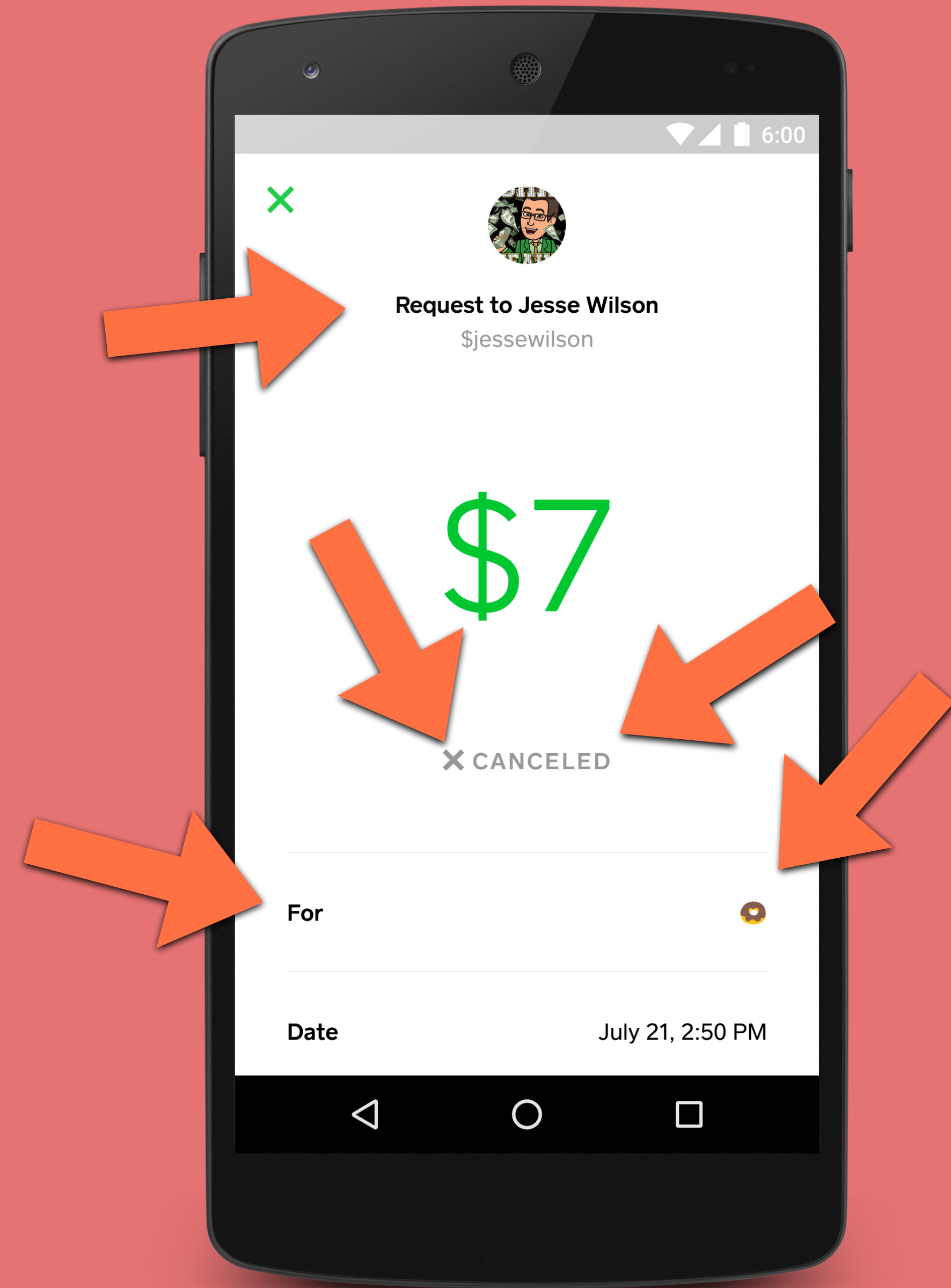
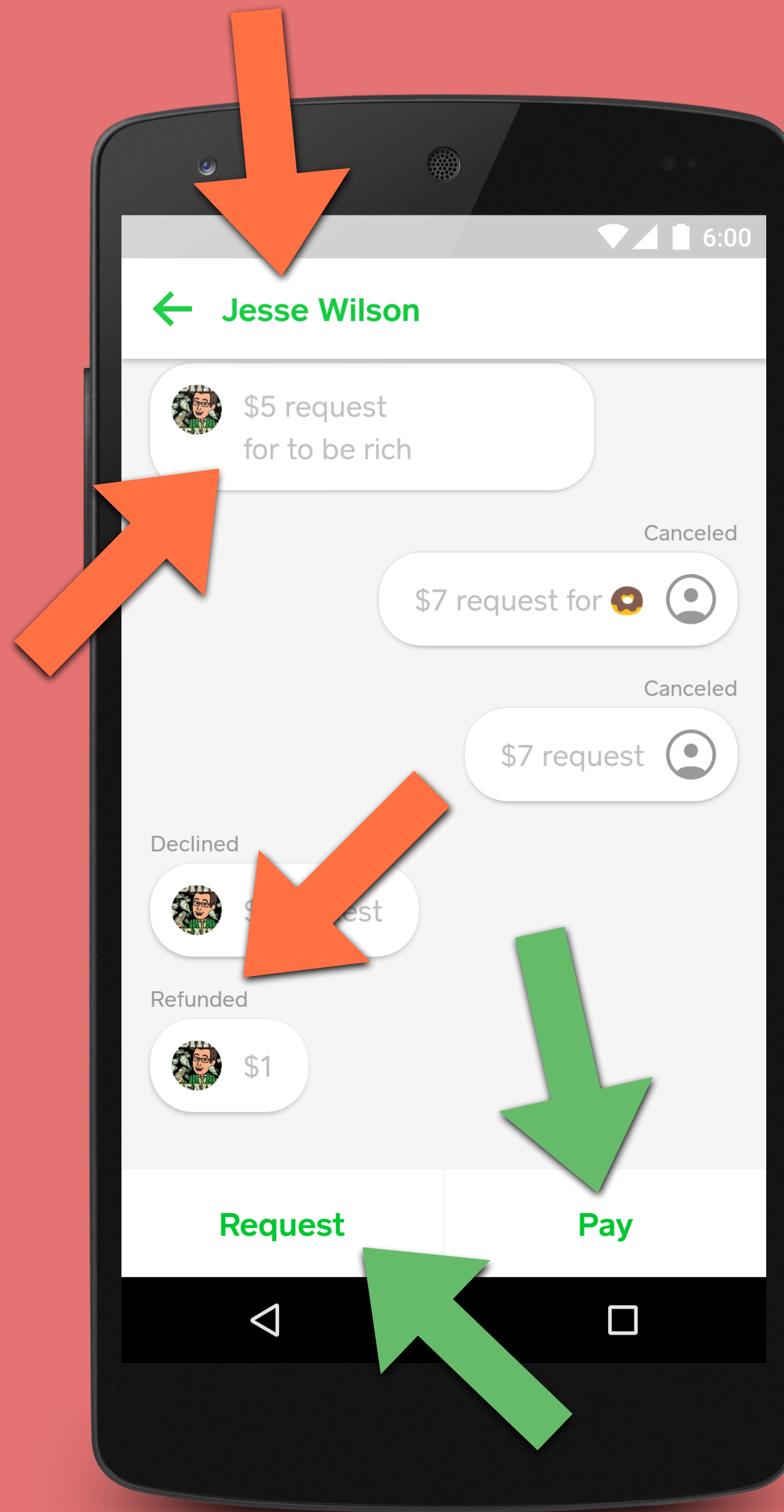
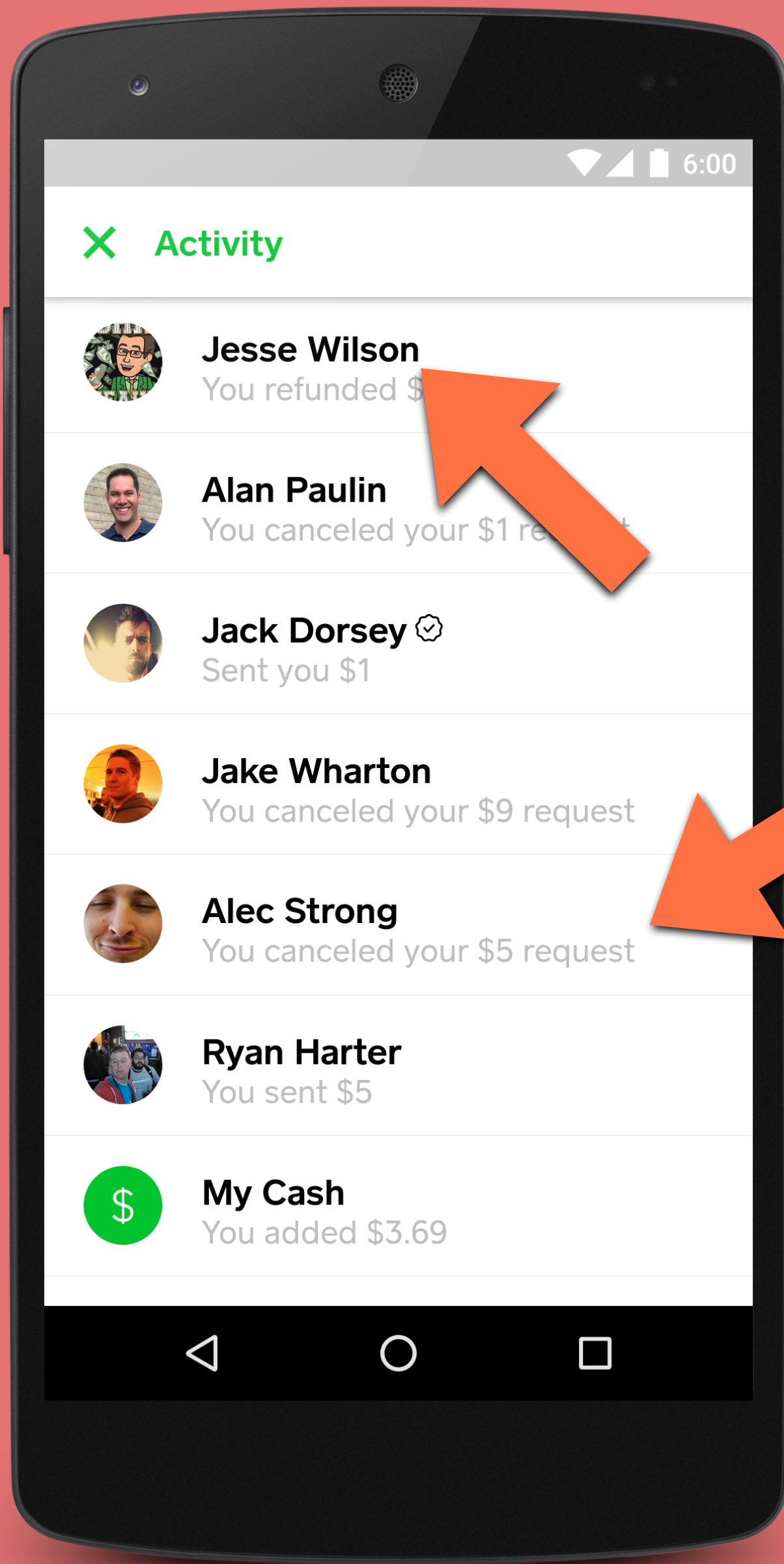


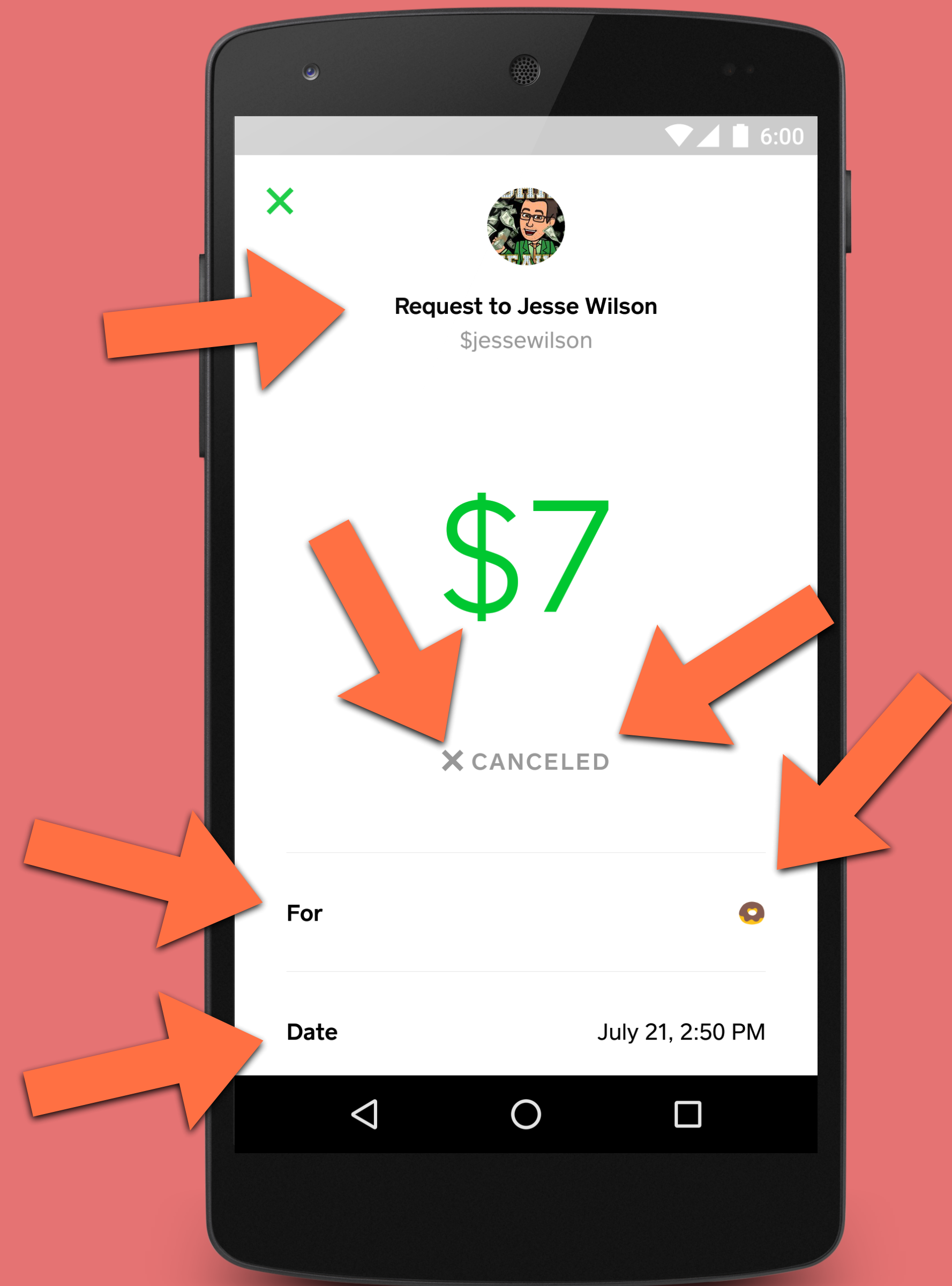
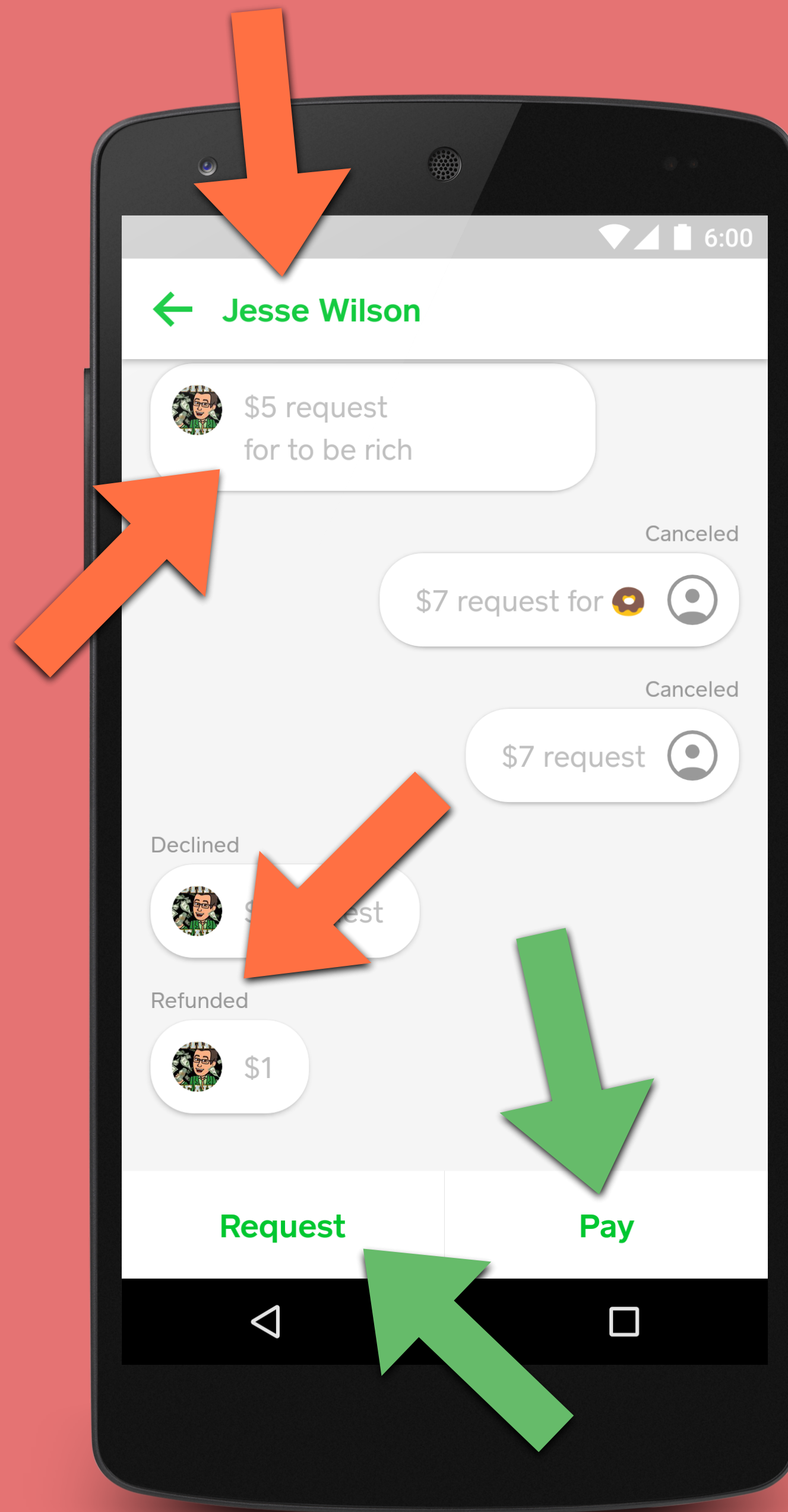
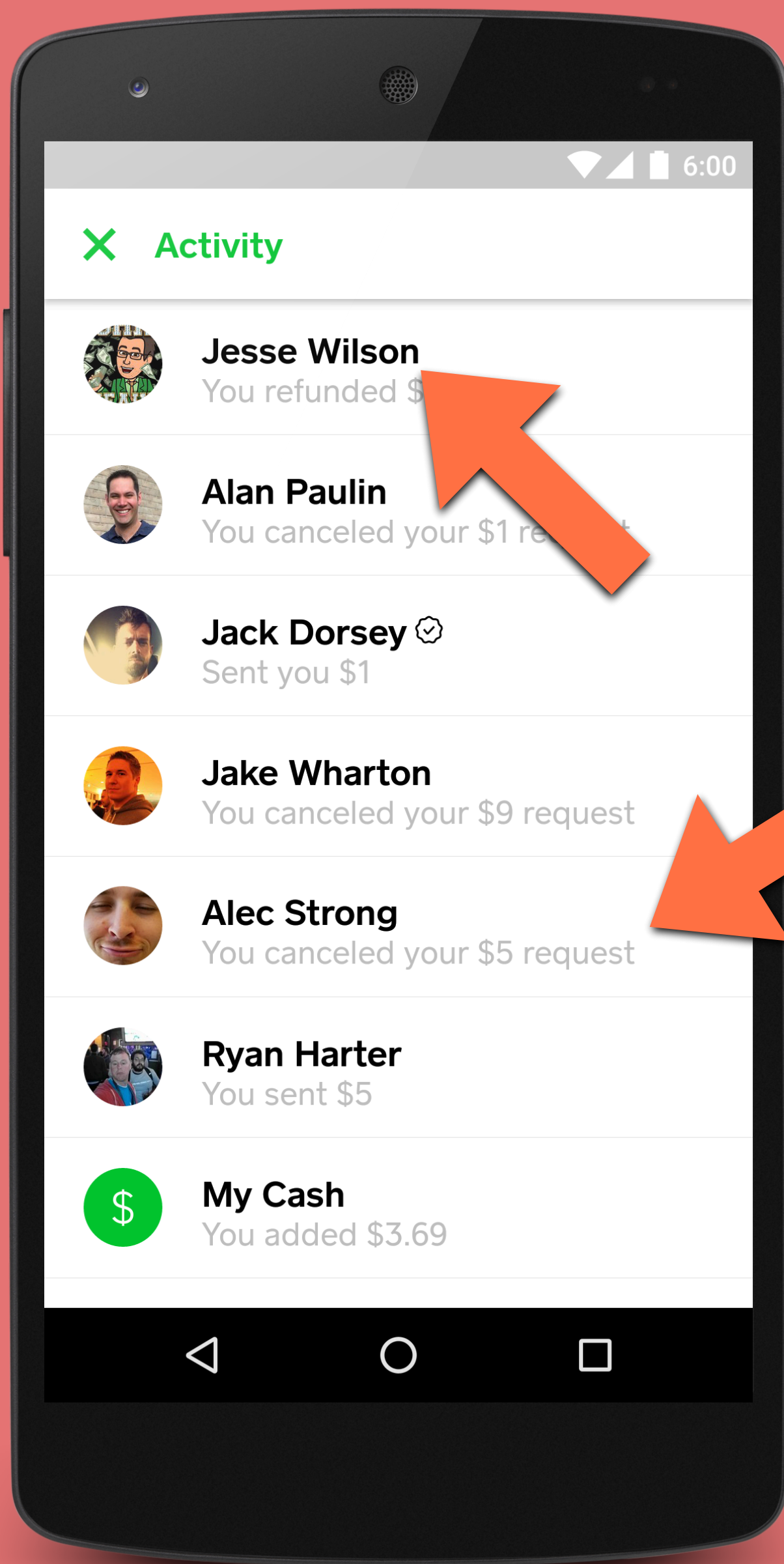


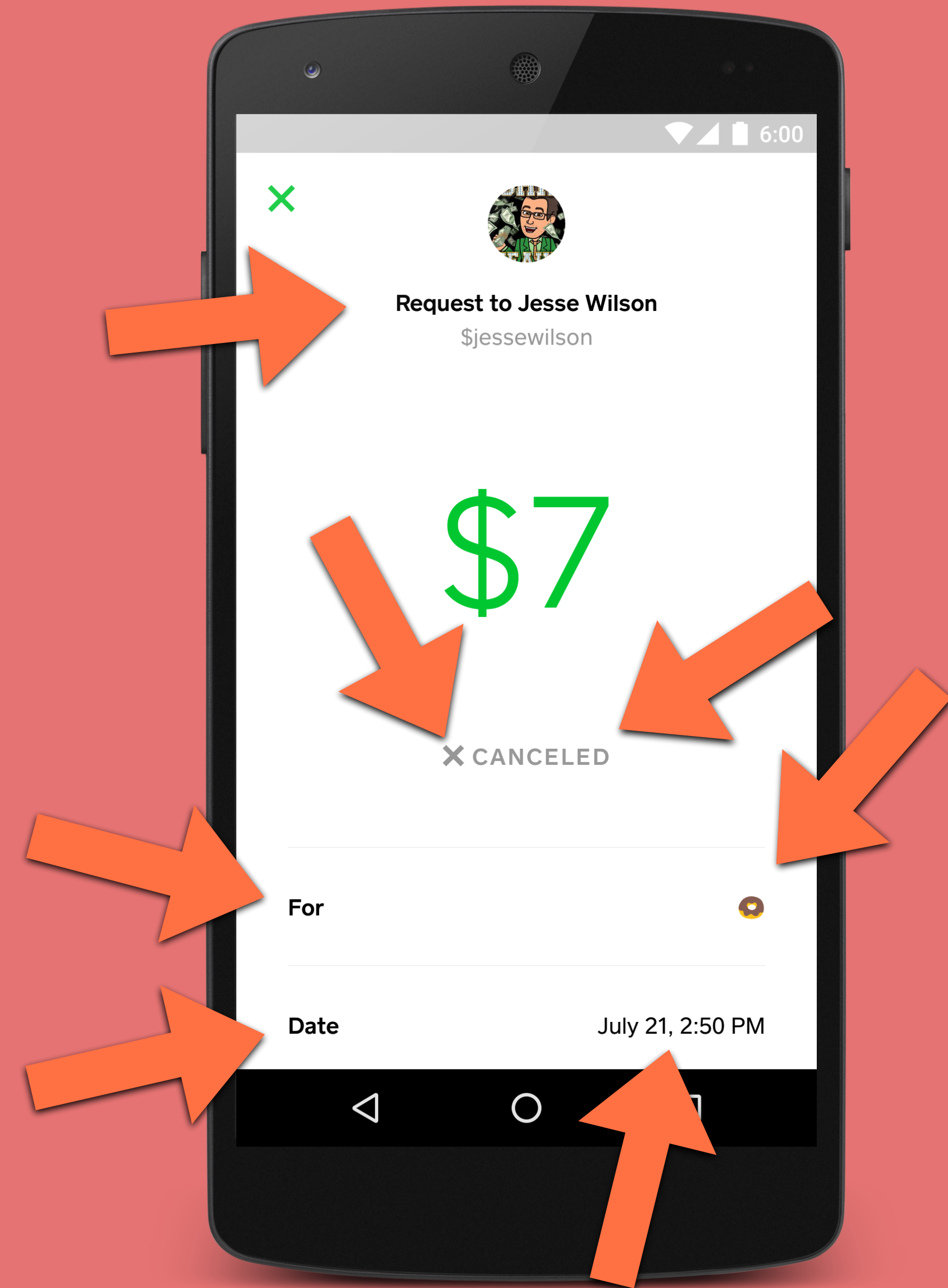
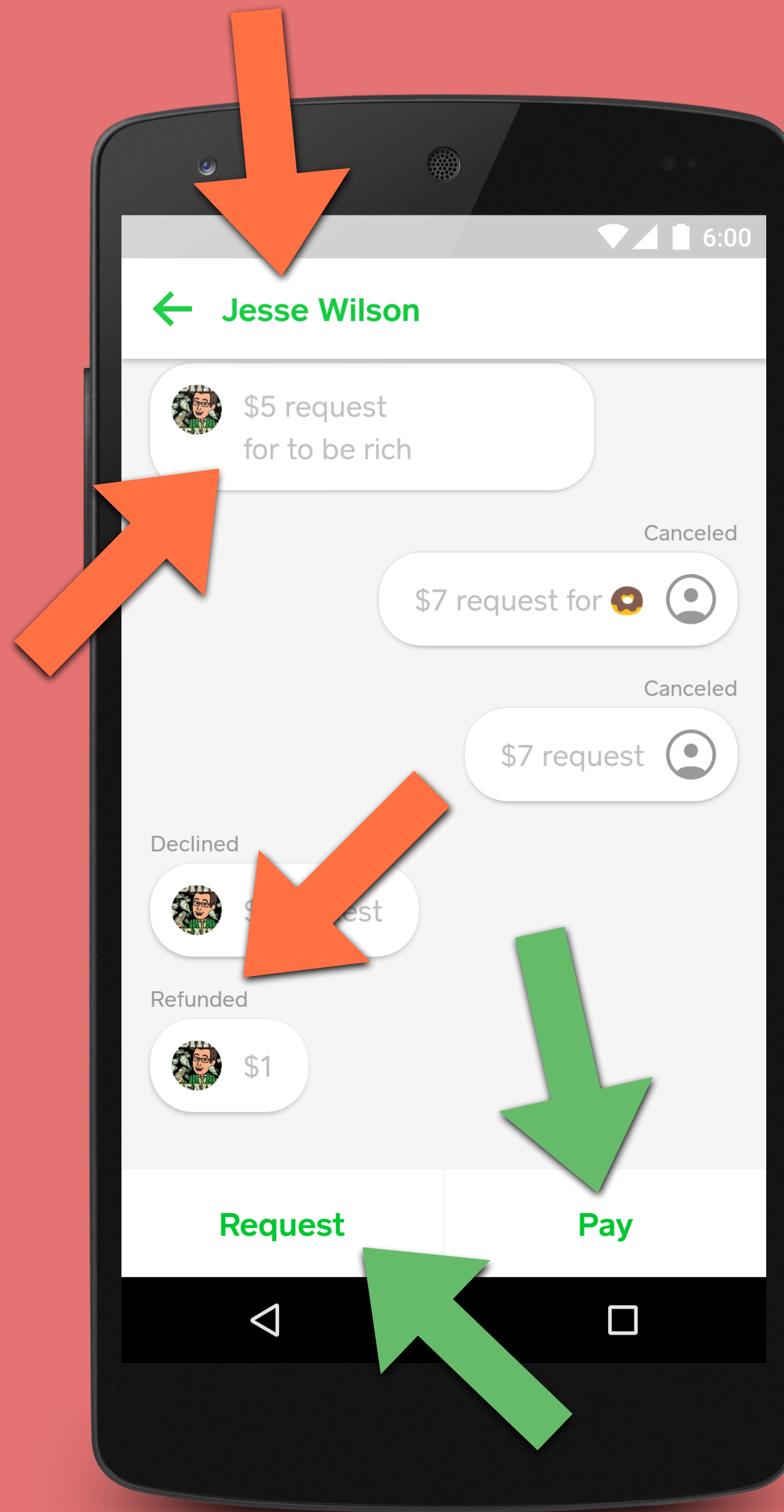
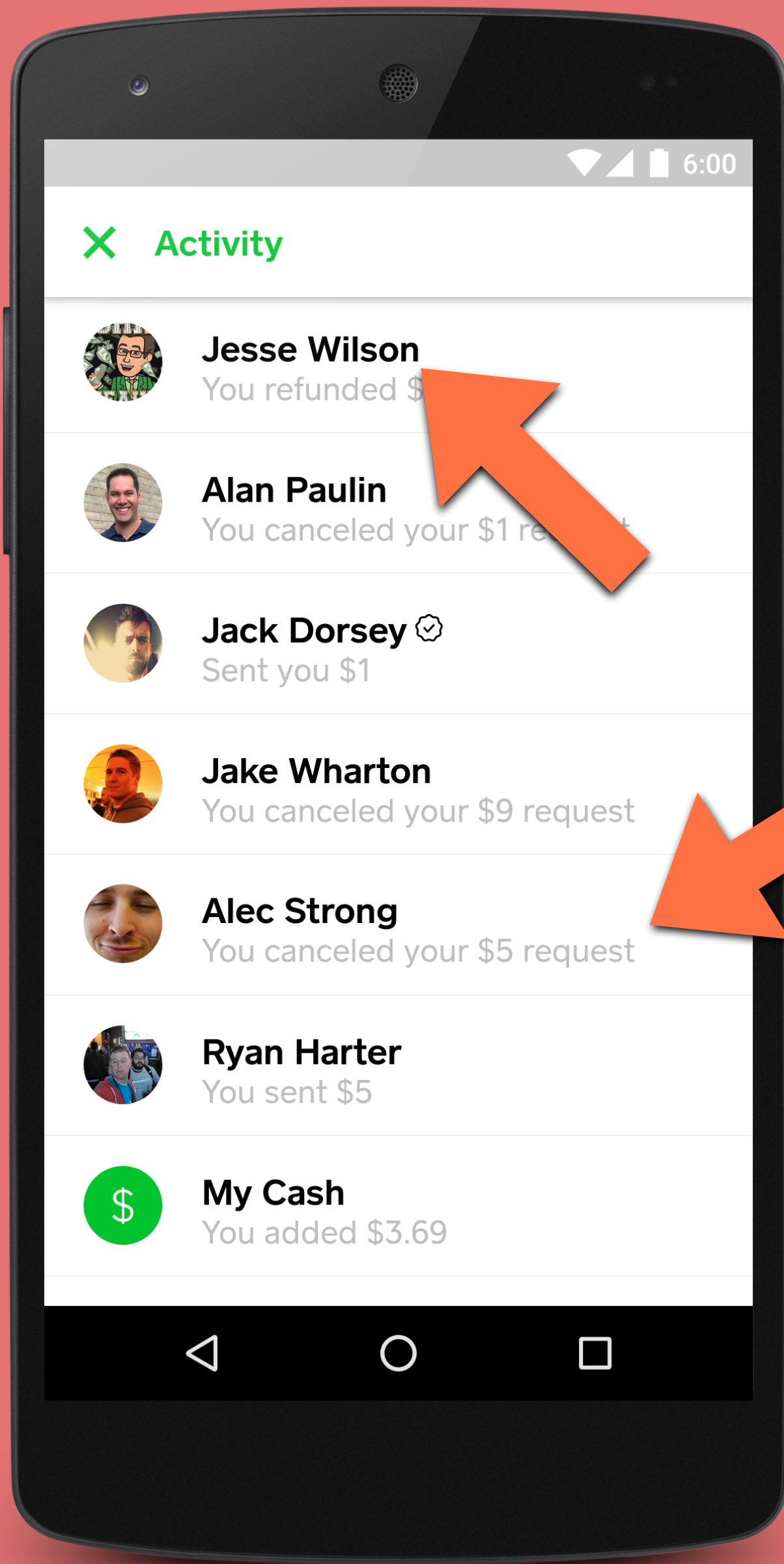








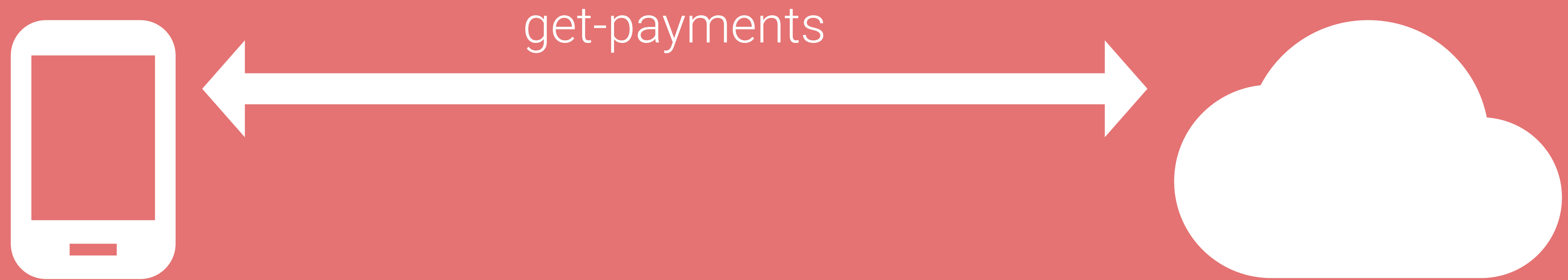




Square Cash: Activity v2?

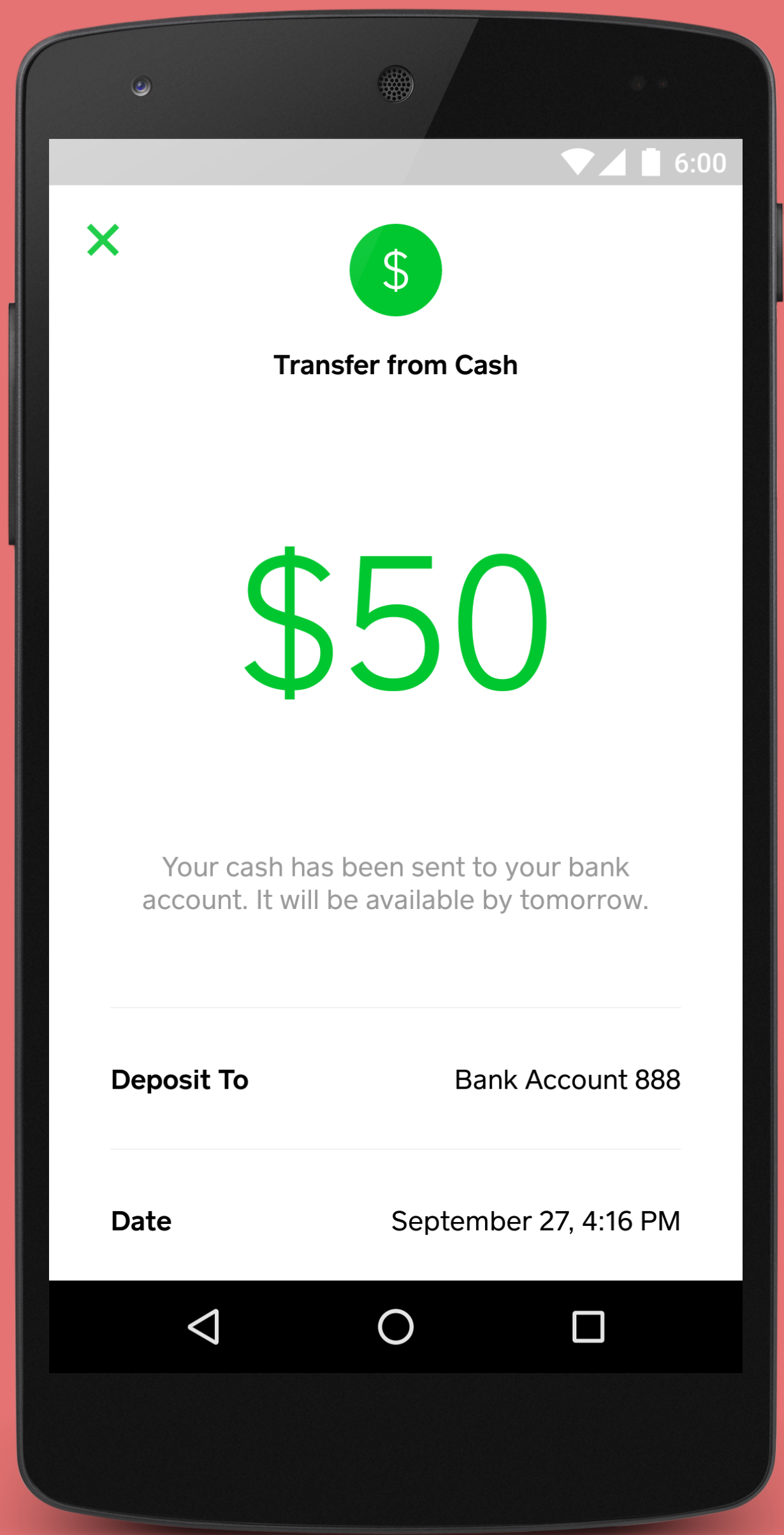


Square Cash: Activity v2?



Square Cash: Activity v2?





\$50

Your cash has been sent to your bank account. It will be available by tomorrow.

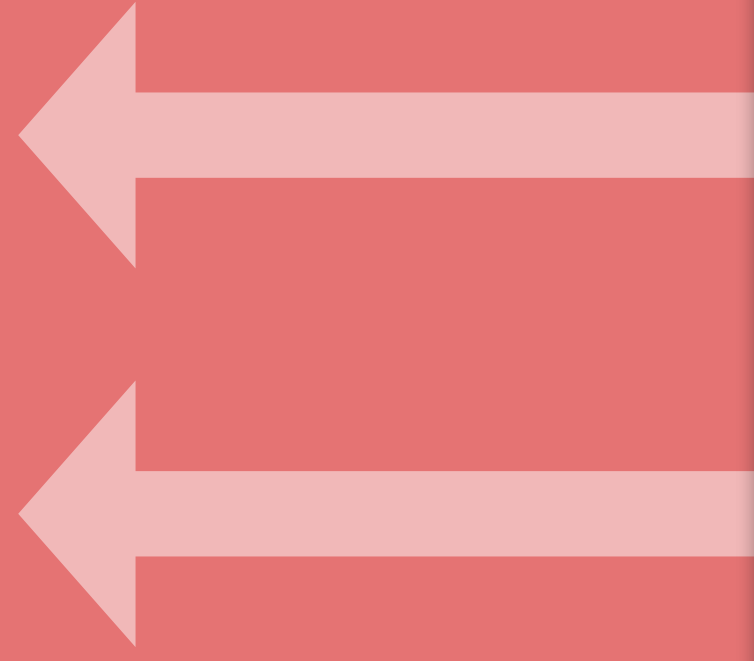
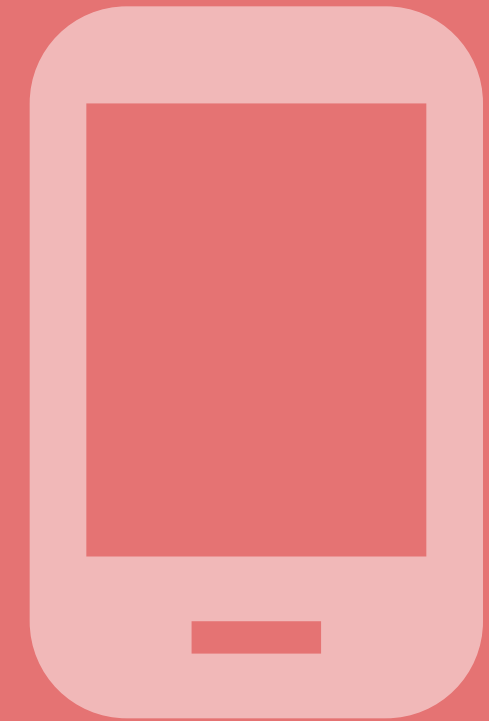
Deposit To

Bank Account 888

Square Cash: Activity v2?



Square Cash: Activity v2?



```
description: {  
  time_1: string_1,  
  time_2: string_2  
}
```



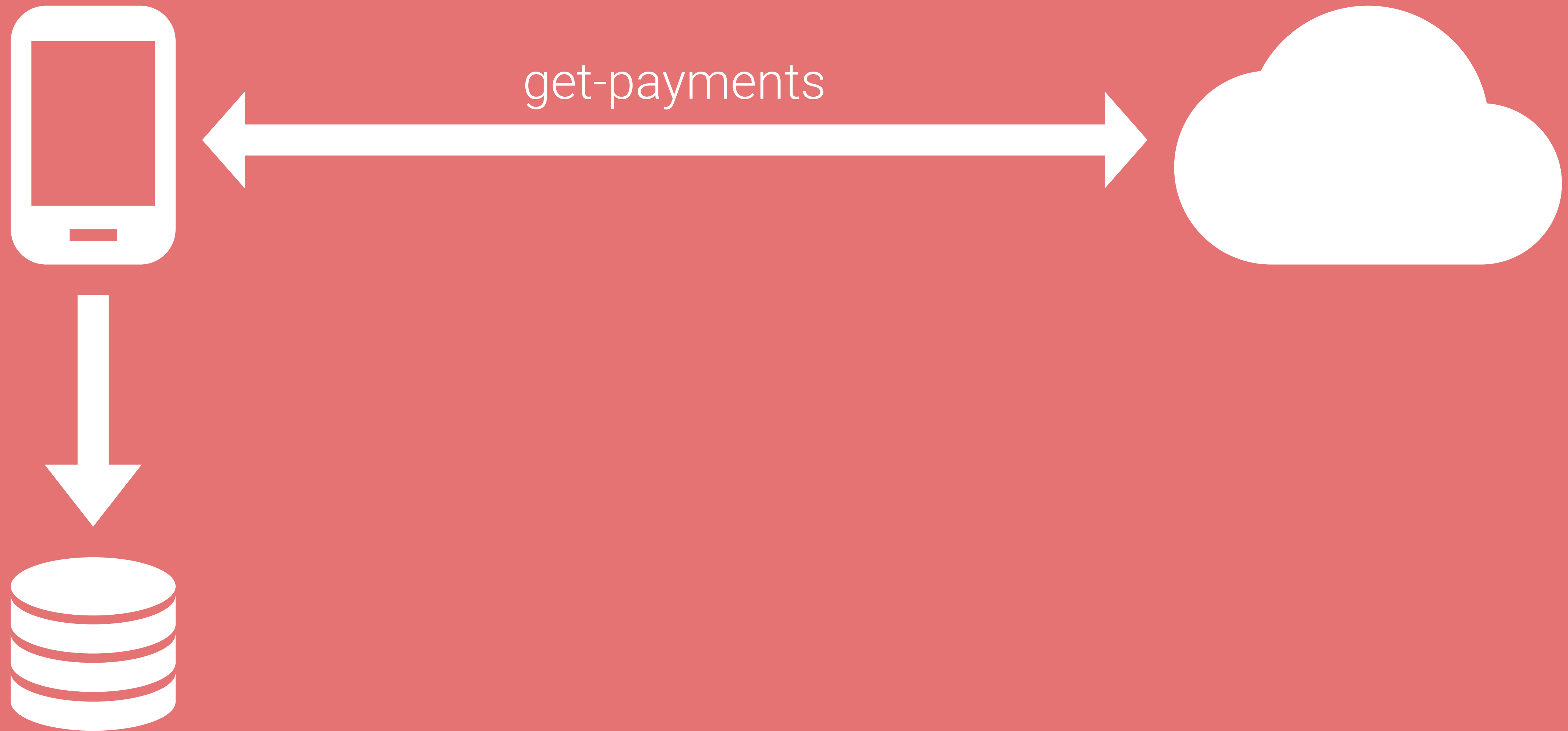
Square Cash: Activity v2!



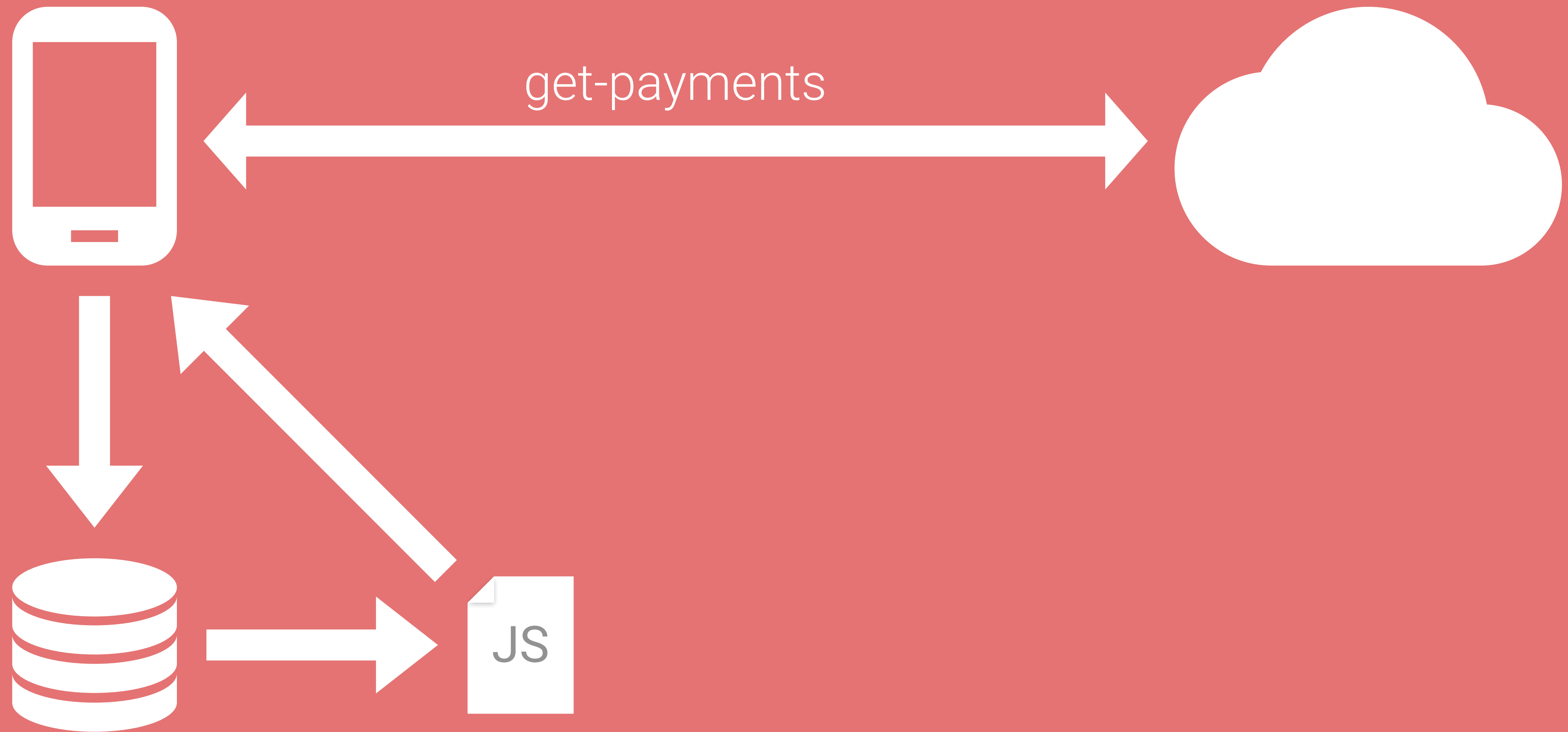
Square Cash: Activity v2!



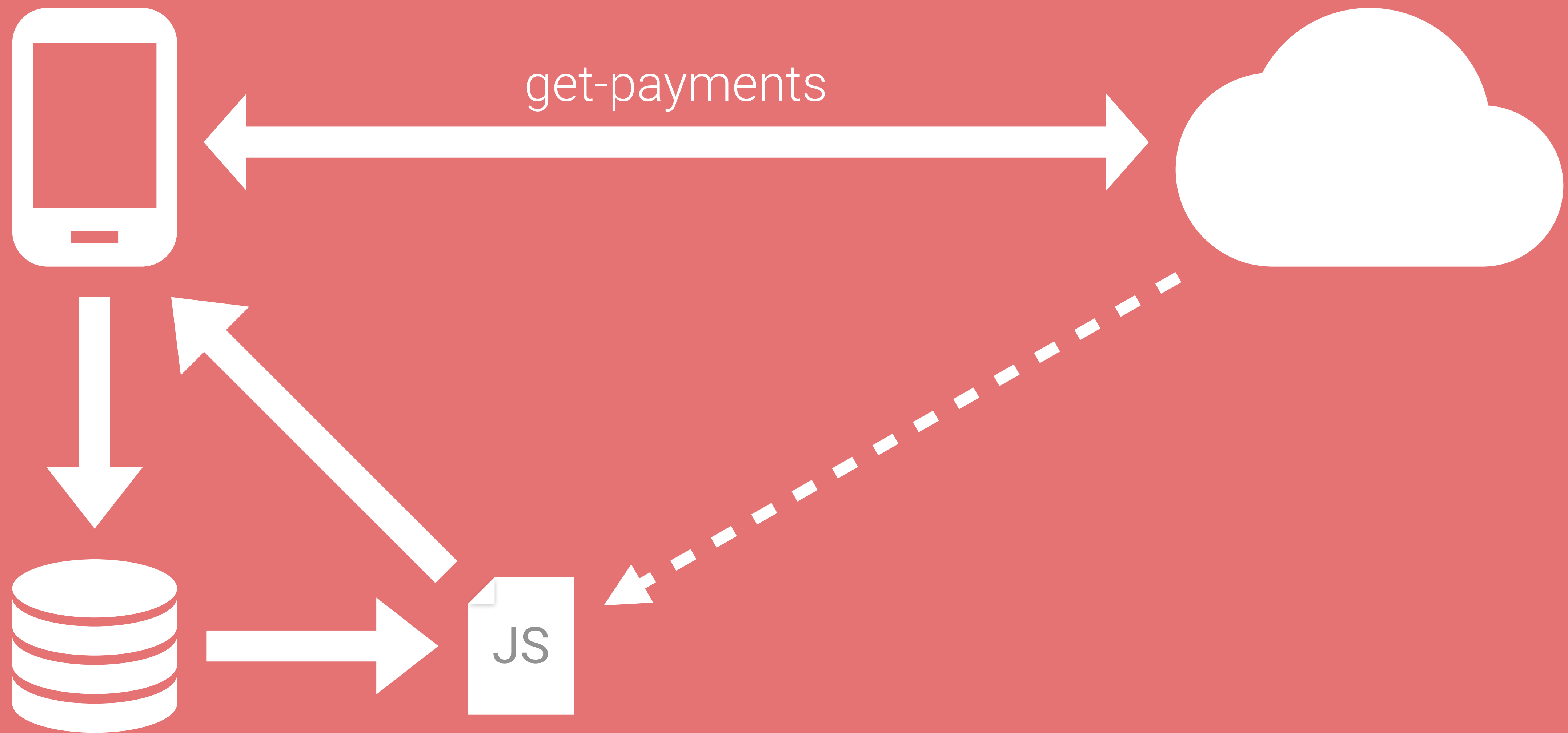
Square Cash: Activity v2!



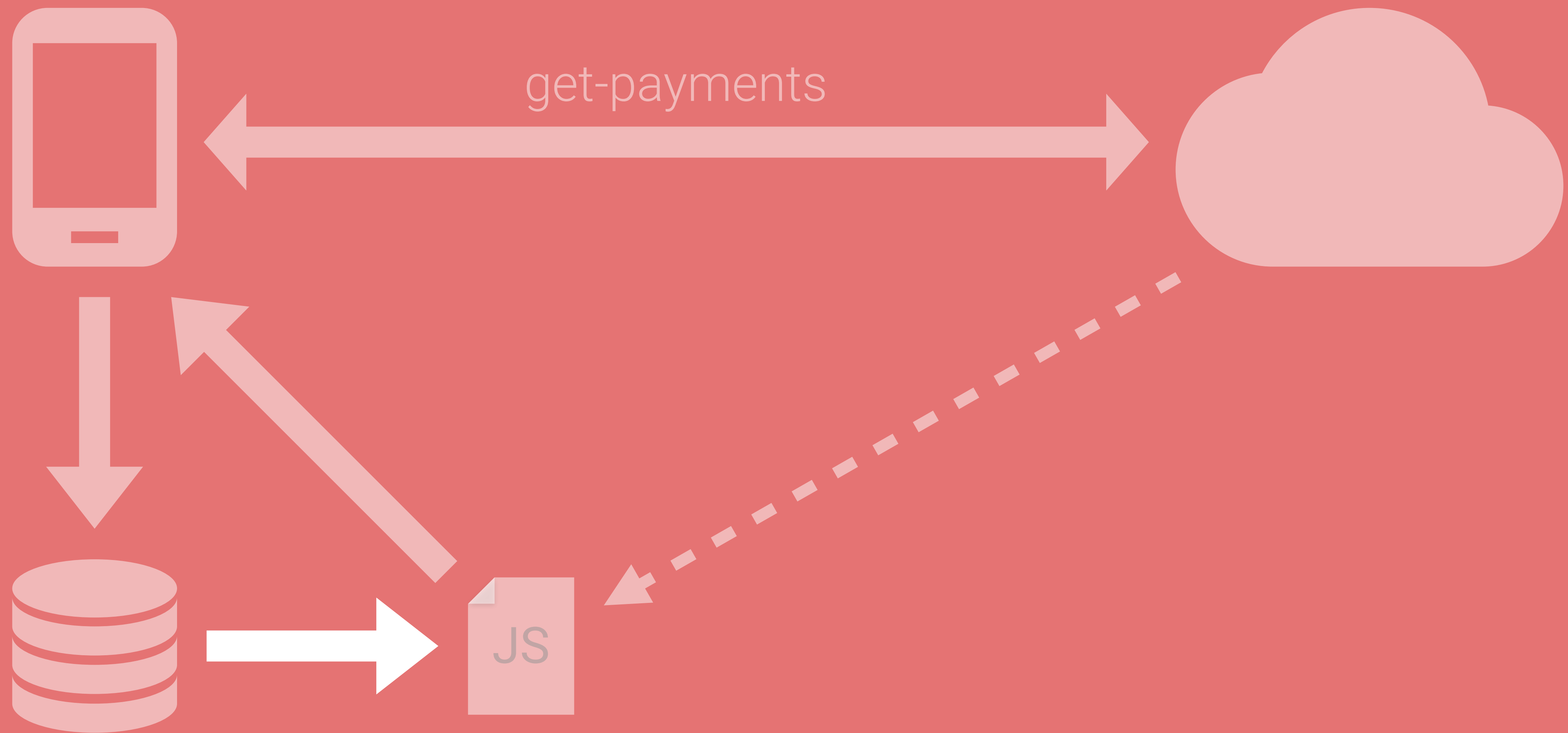
Square Cash: Activity v2!



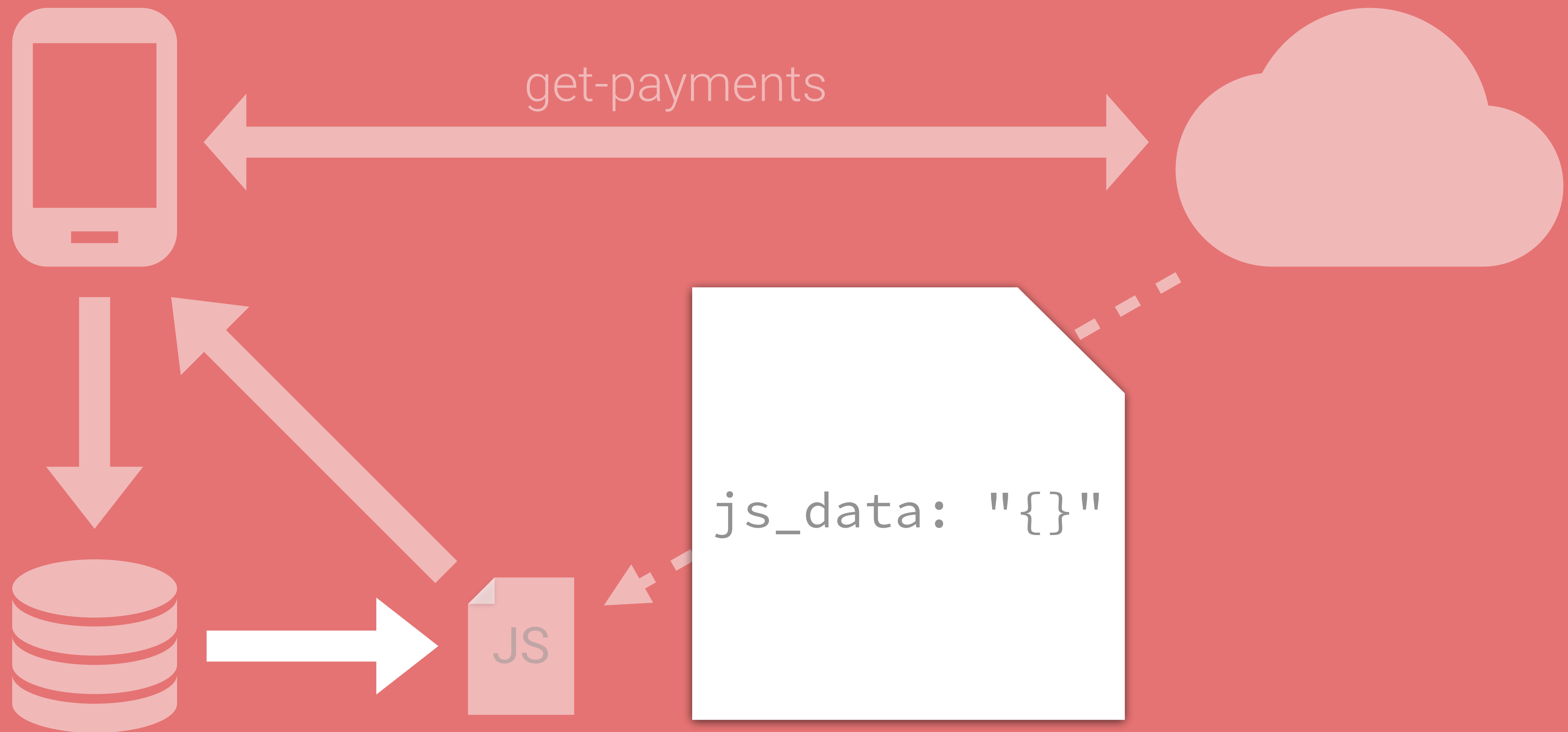
Square Cash: Activity v2!



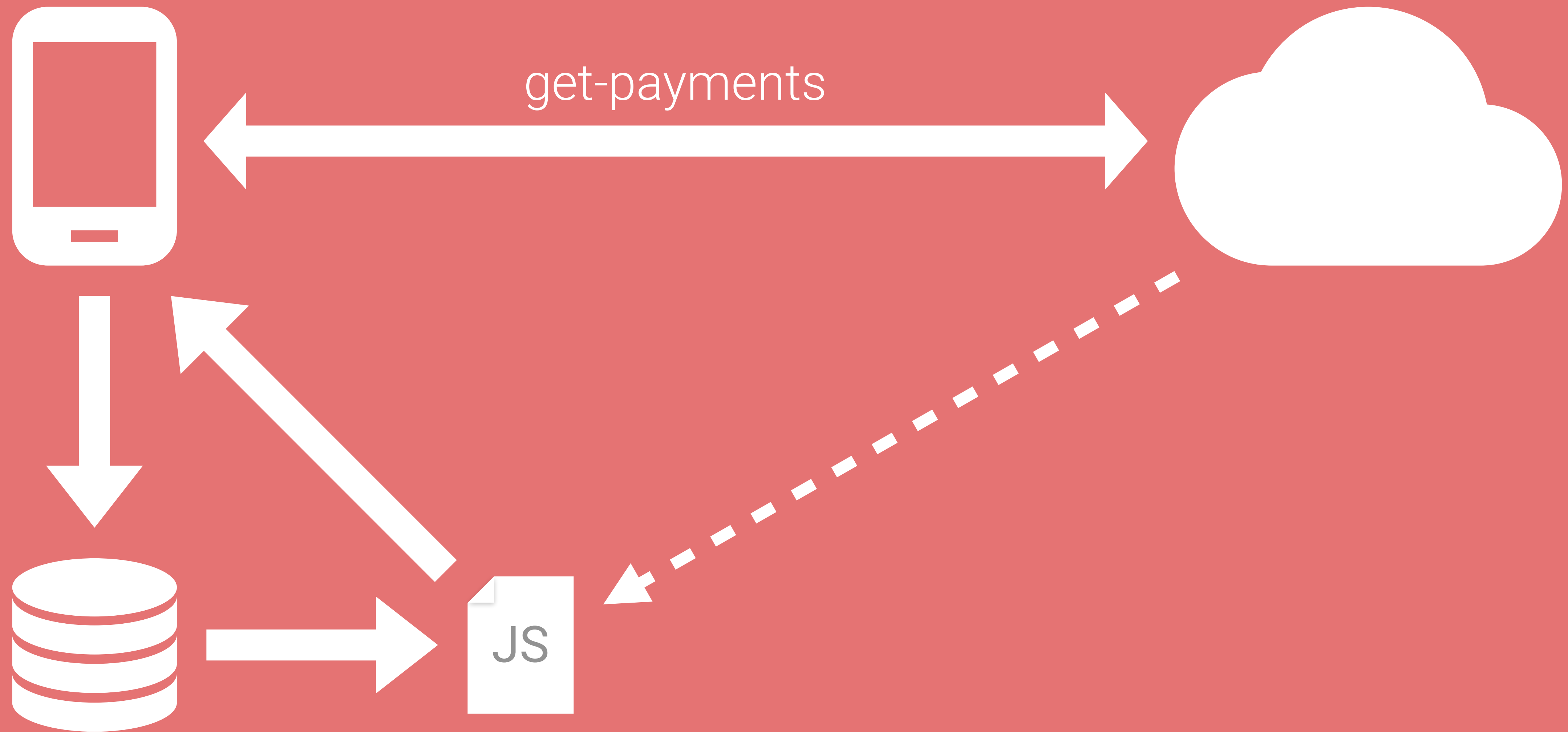
Square Cash: Activity v2!



Square Cash: Activity v2!



Square Cash: Activity v2!



duktape-android pro-tips

duktape-android pro-tips

duktape-android pro-tips

- Duktape is mutable!

duktape-android pro-tips

- Duktape is mutable!
 - Be very careful if you're sharing it between threads

duktape-android pro-tips

- Duktape is mutable!
 - Be very careful if you're sharing it between threads
 - Or don't share it!

duktape-android pro-tips

- Duktape is mutable!
 - Be very careful if you're sharing it between threads
 - Or don't share it!
- ECMAScript E5/E5.1

What is the web's
best feature?

Stealing the Web's Best Feature

▣ Matt Precious
+MatthewPrecious
@mattprec