

```
1: /**
2:  * Sample React Native App
3:  * https://github.com/facebook/react-native
4:  * @flow
5:  */
6:
7: import React, {Component} from 'react';
8: import Firestack from 'react-native-firestack'
9: import moment from 'moment';
10: import {
11:   AppRegistry,
12:   StyleSheet,
13:   Text,
14:   View,
15:   TextInput,
16:   TouchableOpacity,
17:   AppState,
18:   Modal
19: } from 'react-native';
20: import BackgroundTimer from 'react-native-background-timer';
21:
22: import * as firebase from 'firebase';
23: // Initialize Firebase
24: var config = {
25:   apiKey: "AIzaSyDg9Qy5W1YzNK1mmleIvzMBTtxtQIub3kR8",
26:   authDomain: "myreactnative.firebaseio.com",
27:   databaseURL: "https://myreactnative.firebaseio.com",
28:   storageBucket: "myreactnative.appspot.com",
29:   messagingSenderId: "480732941628"
30: };
31: firebase.initializeApp(config);
32: const firestack = new Firestack();
33:
34: export default class l12_firebase extends Component {
35:   constructor(props) {
36:     super(props);
37:     this.database = firebase.database();
38:     this.state = {
39:       chat: '',
40:       chats: [],
41:       userOnline: 0,
42:       modalVisible: false,
43:       name: "Anonymous",
44:       isShowLogin: true,
45:       email: "",
46:       password: ""
47:     }
48:     this.userOnlineRef = this.database.ref('userOnline');
49:     this.chatsRef = this.database.ref('chats');
50:     this.handleAppStateChange = this.handleAppStateChange.bind(this);
51:     this.sendChat = this.sendChat.bind(this);
52:     this.register = this.register.bind(this);
53:     this.login = this.login.bind(this);
54:     this.listeningForAuthChange = this.listeningForAuthChange.bind(this);
55:     AppState.addEventListener('change', this.handleAppStateChange);
56:   }
57:
58:   listeningForAuthChange() {
59:     firestack.auth.listenForAuth((evt) => {
60:       if (!evt.authenticated) {
61:         // There was an error or there is no user
62:         console.log(evt.error)
63:         this.setState({name: 'Anonymous', modalVisible: true});
64:       } else {
65:         // evt.user contains the user details
66:         console.log('User details', evt.user);
67:         this.setState({name: evt.user.email});
68:       }
69:     }).then(() => console.log('Listening for authentication changes'));
70:   }
71:
72:   handleAppStateChange() {
73:     console.log("AppState", AppState.currentState);
74:     if (AppState.currentState == 'active') {
75:       this.getNumberOfUserOnlineOnceAndIncreaseBy1ByTransaction();
76:     } else if (AppState.currentState == 'inactive') {
77:       BackgroundTimer.setTimeout(() => this.decreaseNumberOfUserOnlineByTransaction(), 0);
```

```
78:     }
79:   }
80:
81:   register() {
82:     console.log(this.state.email, this.state.password);
83:     firestore.auth.createUserWithEmail(this.state.email,
84:       this.state.password).then((user) => {
85:       this.setState({isShowLogin: true, modalVisible: false});
86:       console.log("Create user successfully");
87:     }).catch((err) => {
88:       alert("An error occured: " + err.description);
89:       console.log('An error occurred', err);
90:     })
91:   }
92:
93:   login() {
94:     console.log(this.state.email, this.state.password);
95:     firestore.auth.signInWithEmail(this.state.email,
96:       this.state.password).then((user) => {
97:       this.setState({isShowLogin: true, modalVisible: false});
98:       console.log("Login user successfully");
99:     }).catch((err) => {
100:       alert("An error occured: " + err.rawDescription);
101:       console.log('An error occurred', err);
102:     })
103:   }
104:
105:   signOut() {
106:     firestore.auth.signOut().then(res => {
107:       console.log('You have been signed out')
108:     }).catch(err => console.log('Uh oh... something weird happened'));
109:   }
110:
111:   getNumberOfUserOnlineOnceAndIncreaseBy1ByTransaction() {
112:     this.userOnlineRef.transaction(function(currentUserOnline) {
113:       return currentUserOnline + 1;
114:     });
115:   }
116:
117:   decreaseNumberOfUserOnlineByTransaction() {
118:     this.userOnlineRef.transaction(function(currentUserOnline) {
119:       return currentUserOnline > 0
120:         ? currentUserOnline - 1
121:         : 0;
122:     });
123:   }
124:   listeningForNumberOfUserOnline() {
125:     this.userOnlineRef.on('value', (snapshot) => {
126:       console.log("UserOnline change", snapshot.val());
127:       this.setState({userOnline: snapshot.val()});
128:     });
129:   }
130:
131:   listeningForChatChange() {
132:     this.chatsRef.on('value', (snapshot) => {
133:       console.log("Chats change:", snapshot.val());
134:       this.setState({
135:         chats: snapshot.val() || []
136:       });
137:     })
138:   }
139:
140:   componentDidMount() {
141:     this.listeningForNumberOfUserOnline();
142:     this.getNumberOfUserOnlineOnceAndIncreaseBy1ByTransaction();
143:     this.listeningForChatChange();
144:     this.listeningForAuthChange();
145:   }
146:
147:   sendChat() {
148:     console.log(this.state.chat);
149:     this.chatsRef.transaction((chats) => {
150:       if (!chats) {
151:         chats = [];
152:       }
153:       chats.push({name: this.state.name, chat: this.state.chat, when: new Date().getTime()})
154:     });
155:   }
156: }
```

```
154:     this.setState({chat: ""});
155:     return chats;
156:   });
157: }
158:
159: render() {
160:   let authUI;
161:
162:   if (this.state.isShowLogin) {
163:     authUI = (
164:       <View>
165:         <Text style={styles.title}>Login</Text>
166:         <Text>E-mail</Text>
167:         <TextInput keyboardType="email-address" autoCapitalize="none"
168:           style={styles.textInput} value={this.state.email}
169:           onChangeText={(t) => this.setState({email: t})}></TextInput>
170:         <Text>Password</Text>
171:         <TextInput secureTextEntry={true} style={styles.textInput}
172:           value={this.state.password}
173:           onChangeText={(t) => this.setState({password: t})}></TextInput>
174:         <TouchableOpacity style={styles.submitButton} onPress={this.login}>
175:           <Text style={styles.buttonText}>Login</Text>
176:         </TouchableOpacity>
177:         <TouchableOpacity style={styles.secondButton}
178:           onPress={() => this.setState({isShowLogin: false})}>
179:           <Text style={styles.secondButtonText}>Don't have any account?</Text>
180:         </TouchableOpacity>
181:       </View>
182:     )
183:   } else {
184:     authUI = (
185:       <View>
186:         <Text style={styles.title}>Register</Text>
187:         <Text>E-mail</Text>
188:         <TextInput keyboardType="email-address" autoCapitalize="none"
189:           style={styles.textInput} value={this.state.email}
190:           onChangeText={(t) => this.setState({email: t})}></TextInput>
191:         <Text>Password</Text>
192:         <TextInput secureTextEntry={true} style={styles.textInput}
193:           value={this.state.password}
194:           onChangeText={(t) => this.setState({password: t})}></TextInput>
195:         <TouchableOpacity style={styles.submitButton} onPress={this.register}>
196:           <Text style={styles.buttonText}>Register</Text>
197:         </TouchableOpacity>
198:         <TouchableOpacity style={styles.secondButton}
199:           onPress={() => this.setState({isShowLogin: true})}>
200:           <Text style={styles.secondButtonText}>Already have account?</Text>
201:         </TouchableOpacity>
202:       </View>
203:     )
204:   }
205:
206:   let signOut;
207:   if (!this.state.modalVisible) {
208:     signOut = (
209:       <TouchableOpacity style={styles.signOutButton} onPress={this.signOut}>
210:         <Text style={styles.signOutText}>Sign Out</Text>
211:       </TouchableOpacity>
212:     )
213:   }
214:
215:   return (
216:     <View style={styles.container}>
217:       <Modal animationType="slide" transparent={true} visible={this.state.modalVisible}>
218:         <View style={styles.modal}>
219:           <TouchableOpacity style={styles.closeButton} onPress={() => this.setState({modal
Visible: false})}>
220:             <Text>X</Text>
221:           </TouchableOpacity>
222:           {authUI}
223:         </View>
224:       </Modal>
225:       <View style={styles.header}>
226:         <View style={{
227:           flexDirection: 'row'
228:         }}>
229:           <Text style={styles.labelText}>
```

```
230:         Hello {this.state.name}!
231:       </Text>
232:       {signOut}
233:     </View>
234:     <Text style={styles.labelText}>
235:       #User online: {this.state.userOnline}
236:     </Text>
237:   </View>
238:   <View style={styles.content}>
239:     {this.state.chats.map((obj, i) => <View key={i} style={styles.chatContainer}>
240:       <View style={styles.chatMeta}>
241:         <Text style={styles.bold}>{obj.name} || "Anonymous"</Text>
242:         <Text>
243:           ({moment(obj.when).fromNow()})</Text>
244:       </View>
245:       <View style={styles.chat}>
246:         <Text style={styles.chatText}>{obj.chat}</Text>
247:       </View>
248:     </View>)}
249:   </View>
250:   <View style={styles.footer}>
251:     <TextInput style={styles.textInput} value={this.state.chat} onChangeText={(t) => t
his.setState({chat: t})}></TextInput>
252:     <TouchableOpacity style={styles.button} onPress={this.sendChat}>
253:       <Text style={styles.buttonText}>Send</Text>
254:     </TouchableOpacity>
255:   </View>
256: </View>
257: );
258: }
259: }
260:
261: const styles = StyleSheet.create({
262:   chatMeta: {
263:     flexDirection: 'row'
264:   },
265:   chat: {
266:     backgroundColor: 'grey',
267:     borderRadius: 10,
268:     padding: 5
269:   },
270:   chatText: {
271:     color: 'white'
272:   },
273:   container: {
274:     flex: 1,
275:     paddingTop: 20
276:   },
277:   content: {
278:     padding: 10,
279:     flex: 1
280:   },
281:   labelText: {
282:     fontSize: 20
283:   },
284:   header: {
285:     justifyContent: 'center',
286:     alignItems: 'center',
287:     height: 50,
288:     padding: 5,
289:     backgroundColor: '#dddddd'
290:   },
291:   footer: {
292:     height: 50,
293:     backgroundColor: 'yellow',
294:     flexDirection: 'row'
295:   },
296:   textInput: {
297:     flex: 1,
298:     height: 50,
299:     width: 200,
300:     padding: 5,
301:     backgroundColor: 'white',
302:     borderWidth: 1,
303:     borderRadius: 5
304:   },
305:   button: {
```

```
306:     width: 100,
307:     backgroundColor: 'darkblue',
308:     padding: 10,
309:     borderWidth: 1,
310:     borderRadius: 5,
311:     justifyContent: 'center',
312:     alignItems: 'center'
313:   },
314:   buttonText: {
315:     fontSize: 20,
316:     color: 'white'
317:   },
318:   modal: {
319:     height: 280,
320:     width: 300,
321:     marginTop: 150,
322:     padding: 10,
323:     alignSelf: 'center',
324:     backgroundColor: 'lightblue',
325:     margin: 10,
326:     borderRadius: 10,
327:     justifyContent: 'center',
328:     alignItems: 'center'
329:   },
330:   closeButton: {
331:     alignSelf: 'flex-end'
332:   },
333:   secondButtonText: {
334:     color: 'white'
335:   },
336:   secondButton: {
337:     alignSelf: 'center',
338:     backgroundColor: 'gray',
339:     height: 44,
340:     borderRadius: 10,
341:     alignItems: 'center',
342:     justifyContent: 'center',
343:     margin: 10,
344:     padding: 5
345:   },
346:   submitButton: {
347:     alignSelf: 'center',
348:     backgroundColor: 'darkblue',
349:     width: 100,
350:     height: 44,
351:     borderRadius: 10,
352:     alignItems: 'center',
353:     justifyContent: 'center',
354:     margin: 10
355:   },
356:   title: {
357:     fontSize: 20,
358:     fontWeight: 'bold'
359:   },
360:   bold: {
361:     fontWeight: 'bold'
362:   },
363:   signOutButton: {
364:     padding: 5
365:   },
366:   signOutText: {
367:     color: 'red'
368:   }
369: });
370:
371: AppRegistry.registerComponent('l13_authentication', () => l12_firebase);
```

```
1: /**
2:  * Sample React Native App
3:  * https://github.com/facebook/react-native
4:  * @flow
5:  */
6:
7: import React, {Component} from 'react';
8: import Firestack from 'react-native-firestack'
9: import moment from 'moment';
10: import {
11:   AppRegistry,
12:   StyleSheet,
13:   Text,
14:   View,
15:   TextInput,
16:   TouchableOpacity,
17:   AppState,
18:   Modal
19: } from 'react-native';
20: import BackgroundTimer from 'react-native-background-timer';
21:
22: import * as firebase from 'firebase';
23: // Initialize Firebase
24: var config = {
25:   apiKey: "AIzaSyDg9Qy5W1YzNK1mmleIvzMBTtxtQIub3kR8",
26:   authDomain: "myreactnative.firebaseio.com",
27:   databaseURL: "https://myreactnative.firebaseio.com",
28:   storageBucket: "myreactnative.appspot.com",
29:   messagingSenderId: "480732941628"
30: };
31: firebase.initializeApp(config);
32: const firestack = new Firestack();
33:
34: export default class l12_firebase extends Component {
35:   constructor(props) {
36:     super(props);
37:     this.database = firebase.database();
38:     this.state = {
39:       chat: '',
40:       chats: [],
41:       userOnline: 0,
42:       modalVisible: false,
43:       name: "Anonymous",
44:       isShowLogin: true,
45:       email: "",
46:       password: ""
47:     }
48:     this.userOnlineRef = this.database.ref('userOnline');
49:     this.chatsRef = this.database.ref('chats');
50:     this.handleAppStateChange = this.handleAppStateChange.bind(this);
51:     this.sendChat = this.sendChat.bind(this);
52:     this.register = this.register.bind(this);
53:     this.login = this.login.bind(this);
54:     this.listeningForAuthChange = this.listeningForAuthChange.bind(this);
55:     AppState.addEventListener('change', this.handleAppStateChange);
56:   }
57:
58:   listeningForAuthChange() {
59:     firestack.auth.listenForAuth((evt) => {
60:       if (!evt.authenticated) {
61:         // There was an error or there is no user
62:         console.log(evt.error)
63:         this.setState({name: 'Anonymous', modalVisible: true});
64:       } else {
65:         // evt.user contains the user details
66:         console.log('User details', evt.user);
67:         this.setState({name: evt.user.email});
68:       }
69:     }).then(() => console.log('Listening for authentication changes'));
70:   }
71:
72:   handleAppStateChange() {
73:     console.log("AppState", AppState.currentState);
74:     if (AppState.currentState == 'active') {
75:       this.getNumberOfUserOnlineOnceAndIncreaseBy1ByTransaction();
76:     } else if (AppState.currentState == 'inactive') {
77:       BackgroundTimer.setTimeout(() => this.decreaseNumberOfUserOnlineByTransaction(), 0);
```

```
78:     }
79:   }
80:
81:   register() {
82:     console.log(this.state.email, this.state.password);
83:     firestore.auth.createUserWithEmail(this.state.email,
84:       this.state.password).then((user) => {
85:       this.setState({isShowLogin: true, modalVisible: false});
86:       console.log("Create user successfully");
87:     }).catch((err) => {
88:       alert("An error occured: " + err.description);
89:       console.log('An error occurred', err);
90:     })
91:   }
92:
93:   login() {
94:     console.log(this.state.email, this.state.password);
95:     firestore.auth.signInWithEmail(this.state.email,
96:       this.state.password).then((user) => {
97:       this.setState({isShowLogin: true, modalVisible: false});
98:       console.log("Login user successfully");
99:     }).catch((err) => {
100:       alert("An error occured: " + err.rawDescription);
101:       console.log('An error occurred', err);
102:     })
103:   }
104:
105:   signOut() {
106:     firestore.auth.signOut().then(res => {
107:       console.log('You have been signed out')
108:     }).catch(err => console.log('Uh oh... something weird happened'));
109:   }
110:
111:   getNumberOfUserOnlineOnceAndIncreaseBy1ByTransaction() {
112:     this.userOnlineRef.transaction(function(currentUserOnline) {
113:       return currentUserOnline + 1;
114:     });
115:   }
116:
117:   decreaseNumberOfUserOnlineByTransaction() {
118:     this.userOnlineRef.transaction(function(currentUserOnline) {
119:       return currentUserOnline > 0
120:         ? currentUserOnline - 1
121:         : 0;
122:     });
123:   }
124:   listeningForNumberOfUserOnline() {
125:     this.userOnlineRef.on('value', (snapshot) => {
126:       console.log("UserOnline change", snapshot.val());
127:       this.setState({userOnline: snapshot.val()});
128:     });
129:   }
130:
131:   listeningForChatChange() {
132:     this.chatsRef.on('value', (snapshot) => {
133:       console.log("Chats change:", snapshot.val());
134:       this.setState({
135:         chats: snapshot.val() || []
136:       });
137:       this.markAsRead();
138:     })
139:   }
140:
141:   componentDidMount() {
142:     this.listeningForNumberOfUserOnline();
143:     this.getNumberOfUserOnlineOnceAndIncreaseBy1ByTransaction();
144:     this.listeningForChatChange();
145:     this.listeningForAuthChange();
146:   }
147:
148:   sendChat() {
149:     console.log(this.state.chat);
150:     this.chatsRef.transaction((chats) => {
151:       if (!chats) {
152:         chats = [];
153:       }
154:       chats.push({name: this.state.name, chat: this.state.chat, when: new Date().getTime()})
```

```
;
155:     this.setState({chat: ""});
156:     return chats;
157:   });
158: }
159:
160: markAsRead(){
161:   this.chatsRef.transaction((chats) => {
162:     if (!chats) {
163:       chats = [];
164:     }
165:     for (let i in chats){
166:       if (!chats[i].read){
167:         chats[i].read = [];
168:       }
169:       if (chats[i].read.indexOf(this.state.name)<0){
170:         chats[i].read.push(this.state.name);
171:       }
172:     }
173:     this.setState({chat: ""});
174:     return chats;
175:   });
176: }
177:
178: render() {
179:   let authUI;
180:
181:   if (this.state.isShowLogin) {
182:     authUI = (
183:       <View>
184:         <Text style={styles.title}>Login</Text>
185:         <Text>E-mail</Text>
186:         <TextInput keyboardType="email-address" autoCapitalize="none"
187:           style={styles.textInput} value={this.state.email}
188:           onChangeText={(t) => this.setState({email: t})}></TextInput>
189:         <Text>Password</Text>
190:         <TextInput secureTextEntry={true} style={styles.textInput}
191:           value={this.state.password}
192:           onChangeText={(t) => this.setState({password: t})}></TextInput>
193:         <TouchableOpacity style={styles.submitButton} onPress={this.login}>
194:           <Text style={styles.buttonText}>Login</Text>
195:         </TouchableOpacity>
196:         <TouchableOpacity style={styles.secondButton}
197:           onPress={() => this.setState({isShowLogin: false})}>
198:           <Text style={styles.secondButtonText}>Don't have any account?</Text>
199:         </TouchableOpacity>
200:       </View>
201:     )
202:   } else {
203:     authUI = (
204:       <View>
205:         <Text style={styles.title}>Register</Text>
206:         <Text>E-mail</Text>
207:         <TextInput keyboardType="email-address" autoCapitalize="none"
208:           style={styles.textInput} value={this.state.email}
209:           onChangeText={(t) => this.setState({email: t})}></TextInput>
210:         <Text>Password</Text>
211:         <TextInput secureTextEntry={true} style={styles.textInput}
212:           value={this.state.password}
213:           onChangeText={(t) => this.setState({password: t})}></TextInput>
214:         <TouchableOpacity style={styles.submitButton} onPress={this.register}>
215:           <Text style={styles.buttonText}>Register</Text>
216:         </TouchableOpacity>
217:         <TouchableOpacity style={styles.secondButton}
218:           onPress={() => this.setState({isShowLogin: true})}>
219:           <Text style={styles.secondButtonText}>Already have account?</Text>
220:         </TouchableOpacity>
221:       </View>
222:     )
223:   }
224:
225:   let signOut;
226:   if (!this.state.modalVisible) {
227:     signOut = (
228:       <TouchableOpacity style={styles.signOutButton} onPress={this.signOut}>
229:         <Text style={styles.signOutText}>Sign Out</Text>
230:       </TouchableOpacity>
```



```

231:     )
232:   }
233:
234:   return (
235:     <View style={styles.container}>
236:       <Modal animationType={"slide"} transparent={true} visible={this.state.modalVisible}>
237:         <View style={styles.modal}>
238:           <TouchableOpacity style={styles.closeButton} onPress={() => this.setState({modal
Visible: false})}>
239:             <Text>X</Text>
240:           </TouchableOpacity>
241:           {authUI}
242:         </View>
243:       </Modal>
244:       <View style={styles.header}>
245:         <View style={{
246:           flexDirection: 'row'
247:         }}>
248:           <Text style={styles.labelText}>
249:             Hello {this.state.name}!
250:           </Text>
251:           {signOut}
252:         </View>
253:         <Text style={styles.labelText}>
254:           #User online: {this.state.userOnline}
255:         </Text>
256:       </View>
257:       <View style={styles.content}>
258:         {this.state.chats.map((obj, i) => <View key={i} style={styles.chatContainer}>
259:           <View style={styles.chatMeta}>
260:             <Text style={styles.bold}>{obj.name || "Anonymous"}</Text>
261:             <Text>
262:               ({moment(obj.when).fromNow()}</Text>
263:             <Text>
264:               Read by: {(obj.read || []).length}
265:             </Text>
266:           </View>
267:           <View style={styles.chat}>
268:             <Text style={styles.chatText}>{obj.chat}</Text>
269:           </View>
270:         </View>)}
271:       </View>
272:       <View style={styles.footer}>
273:         <TextInput style={styles.textInput} value={this.state.chat} onChangeText={(t) => t
his.setState({chat: t})}></TextInput>
274:         <TouchableOpacity style={styles.button} onPress={this.sendChat}>
275:           <Text style={styles.buttonText}>Send</Text>
276:         </TouchableOpacity>
277:       </View>
278:     </View>
279:   );
280: }
281: }
282:
283: const styles = StyleSheet.create({
284:   chatMeta: {
285:     flexDirection: 'row'
286:   },
287:   chat: {
288:     backgroundColor: 'grey',
289:     borderRadius: 10,
290:     padding: 5
291:   },
292:   chatText: {
293:     color: 'white'
294:   },
295:   container: {
296:     flex: 1,
297:     paddingTop: 20
298:   },
299:   content: {
300:     padding: 10,
301:     flex: 1
302:   },
303:   labelText: {
304:     fontSize: 20
305:   },

```

```
306:   header: {
307:     justifyContent: 'center',
308:     alignItems: 'center',
309:     height: 50,
310:     padding: 5,
311:     backgroundColor: '#dddddd'
312:   },
313:   footer: {
314:     height: 50,
315:     backgroundColor: 'yellow',
316:     flexDirection: 'row'
317:   },
318:   textInput: {
319:     flex: 1,
320:     height: 50,
321:     width: 200,
322:     padding: 5,
323:     backgroundColor: 'white',
324:     borderWidth: 1,
325:     borderRadius: 5
326:   },
327:   button: {
328:     width: 100,
329:     backgroundColor: 'darkblue',
330:     padding: 10,
331:     borderWidth: 1,
332:     borderRadius: 5,
333:     justifyContent: 'center',
334:     alignItems: 'center'
335:   },
336:   buttonText: {
337:     fontSize: 20,
338:     color: 'white'
339:   },
340:   modal: {
341:     height: 280,
342:     width: 300,
343:     marginTop: 150,
344:     padding: 10,
345:     alignSelf: 'center',
346:     backgroundColor: 'lightblue',
347:     margin: 10,
348:     borderRadius: 10,
349:     justifyContent: 'center',
350:     alignItems: 'center'
351:   },
352:   closeButton: {
353:     alignSelf: 'flex-end'
354:   },
355:   secondButtonText: {
356:     color: 'white'
357:   },
358:   secondButton: {
359:     alignSelf: 'center',
360:     backgroundColor: 'gray',
361:     height: 44,
362:     borderRadius: 10,
363:     alignItems: 'center',
364:     justifyContent: 'center',
365:     margin: 10,
366:     padding: 5
367:   },
368:   submitButton: {
369:     alignSelf: 'center',
370:     backgroundColor: 'darkblue',
371:     width: 100,
372:     height: 44,
373:     borderRadius: 10,
374:     alignItems: 'center',
375:     justifyContent: 'center',
376:     margin: 10
377:   },
378:   title: {
379:     fontSize: 20,
380:     fontWeight: 'bold'
381:   },
382:   bold: {
```

```
383:     fontWeight: 'bold'
384:   },
385:   signOutButton: {
386:     padding: 5
387:   },
388:   signOutText: {
389:     color: 'red'
390:   }
391: });
392:
393: AppRegistry.registerComponent('l13_authentication', () => l12_firebase);
```