

```

import React, { Component } from 'react';
import {
  StyleSheet,
  Text,
  View,
  Image,
  ActivityIndicator,
  FlatList,
  TextInput
} from 'react-native';
import { Pages, Indicator } from 'react-native-pages';
import * as ImagePicker from 'expo-image-picker';
import firebase from 'firebase';
import '@firebase/firestore'
import 'firebase/storage'
var uuid = require('react-native-uuid');

import ButtonComponent from '../components/ButtonComponent';
import FormInput from '../components/FormInput';
import { TouchableOpacity } from 'react-native-gesture-handler';
import { Picker } from 'react-native-form-idable';
import { Icon } from 'react-native-elements';
import Toast from 'react-native-simple-toast';

class FileClaim extends Component {
  constructor(props) {
    super(props);

    this.state = {
      description: null,
      quoteImage: null,
      qID: null,
      reportImage: null,
      rID: null,
      contacts: [],
      contact_role: '',
      contact_name: '',
      contact_phone: '',
      contact_email: '',
      validCount: 0,
      claimAmount: 0
    }
  }

  async componentDidMount(){

  }

  //upload quote image
  async quoteImage(type){
    let result = null
    switch (type) {
      case 'camera':
        //camera upload
        result = await this._showCamera()
        break;
      case 'cameraroll':
        //gallery upload

```

```

        result = await this._showCameraRoll()
        break;
      default:
        break;
    }

    if (!result.cancelled) {
      this.setState({ quoteImage: result.uri, qID: uuid.v4(), loading: false })
    }
  }

  //upload report quote
  async reportImage(type){
    let result = null
    switch (type) {
      case 'camera':
        //camera upload
        result = await this._showCamera()
        break;
      case 'cameraroll':
        //gallery upload
        result = await this._showCameraRoll()
        break;
      default:
        break;
    }

    if (!result.cancelled) {
      this.setState({ reportImage: result.uri, rID: uuid.v4(), loading: false });
    }
  }

  //display inbuilt gallery
  async _showCameraRoll(){
    if(!ImagePicker.getCameraRollPermissionsAsync()){
      ImagePicker.requestCameraRollPermissionsAsync()
    } else{
      let result = await ImagePicker.launchImageLibraryAsync({
        mediaTypes: ImagePicker.MediaTypeOptions.All,
        allowsEditing: true,
        aspect: [4, 3],
        quality: 1
      })

      return result
    }
  }

  //display camera
  async _showCamera(){
    if(!ImagePicker.getCameraPermissionsAsync()){
      ImagePicker.requestCameraPermissionsAsync()
    } else{
      let result = await ImagePicker.launchCameraAsync({
        quality: 1
      });
    }
  }

```

```

        return result
    }
}

//convert local path to javascript blob
uriToBlob = (uri) => {
    return new Promise((resolve, reject) => {
        const xhr = new XMLHttpRequest();
        xhr.onload = function() {
            // return the blob
            resolve(xhr.response);
        };

        xhr.onerror = function() {
            // something went wrong
            reject(new Error('uriToBlob failed'));
        };

        xhr.responseType = 'blob';
        xhr.open('GET', uri, true);

        xhr.send(null);
    });
}

//upload images to storage
uploadToFirebase = (blob, id) => {
    return new Promise((resolve, reject)=>{
        var storageRef = firebase.storage().ref()
        storageRef.child(`uploads/${firebase.auth().currentUser.uid}
/${id}.jpeg`).put(blob, {
            contentType: 'image/jpeg'
        }).then((snapshot)=>{
            blob.close();
            resolve(snapshot);
        }).catch((error)=>{
            reject(error);
        });
    });
}

//build contact list
appendInput(){
    const data = {
        name: this.state.contact_name,
        email: this.state.contact_email,
        phone: this.state.contact_phone,
        role: this.state.contact_role
    }
    this.setState(prevState => ({ contacts: prevState.contacts.concat([data]) })))
}

//display contact within contact list
renderContact(item, index){
    console.log(item)
    return (
        <View style={{paddingLeft: '10%', borderBottomWidth: 1, borderBottomColor:
'#2E6CB5', paddingBottom: 10}}>
            <Text style={{fontSize: 18, fontWeight: 'bold'}}>{item.role}</Text>

```

```

        <Text style={{fontSize: 18}}>Name: {item.name}</Text>
        <Text style={{fontSize: 18}}>Phone Number: {item.phone}</Text>
        <Text style={{fontSize: 18}}>Email: {item.email}</Text>
        <TouchableOpacity
          onPress={() => this.deleteContact(index)}
        >
          <Text style={{textAlign: 'right', paddingRight: 20, color:
'#2E6CB5'}}>Remove Contact</Text>
        </TouchableOpacity>
      </View>
    )
  }

  //remove contact
  deleteContact(index){
    var array = [...this.state.contacts]
    array.splice(index, 1)
    this.setState({contacts: array})
  }

  //EVENT: firebase upload
  submitClaim(){
    const { currentUser } = firebase.auth()
    const amount = Number(this.state.claimAmount)

    //check all required fields are filled
    if(this.state.contacts.length === 0
      || this.state.description === null
      || this.state.quoteImage.uri === null
      || this.state.reportImage === null){
      Toast.show('Please complete all tasks before submitting')
    } else if(amount < 0 || amount == undefined){
      Toast.show('Please enter a valid amount')
    } else {
      const day = new Date().getDate()
      const month = new Date().getMonth()
      const year = new Date().getFullYear()
      const date = day + '/' + month + '/' + year

      this.uriToBlob(this.state.quoteImage)
        .then(blob => {
          this.uploadToFirebase(blob, this.state.qID)
        }).then(snapshot => {
          console.log('Quote image uploaded')
        }).catch(e => console.log(e))

      this.uriToBlob(this.state.reportImage)
        .then(blob => {
          this.uploadToFirebase(blob, this.state.rID)
        }).then(snapshot => {
          console.log('Report image uploaded')
        }).catch(e => console.log(e))

      const data = {
        complete: false,
        contact_list: this.state.contacts,
        description: this.state.description,
        quote_image_id: this.state.qID,

```

```

        report_image_id: this.state.rID,
        date_submitted: date,
        date_reviewed: '',
        outcome: '',
        amount: amount
    }

    //upload claim to /users/<uid>/claims
    firebase.database().ref(`users/${currentUser.uid}/claims/`)
        .push(data)

    //navigate to claim filed screen
    this.props.navigation.navigate('ClaimUploaded')
  }
}

render() {
  let { quoteImage, loading, reportImage } = this.state;

  return(
    <Pages
      indicatorColor='#2E6CB5'
    >
      <View style={styles.container}>
        <Text style={styles.logo}><Related Contact Info</Text>
        <View style={styles.form}>
          <Picker
            name="role"
            placeholder="Please select contacts role"
            formStyles={pickerStyle}
            onChangeText={(value) => this.setState({contact_role:
value}})
          >
            <Picker.item label='Please select contacts role'
value='empty' />
            <Picker.Item label="Witness" value="Witness" />
            <Picker.Item label="Police Officer" value="Police Officer"
/>
          </Picker>

          <FormInput
            icon='user'
            type='antdesign'
            placeholder='Full name'
            onChangeText={value => this.setState({contact_name: value})}
          />

          <FormInput
            icon='phone'
            type='antdesign'
            placeholder='Phone Number'
            onChangeText={value => this.setState({contact_phone:
value}})
          />

          <FormInput
            icon='mail'

```

```

        type='antdesign'
        placeholder='Email Address'
        onChangeText={value => this.setState({contact_email:
value}})}

        />

        <TouchableOpacity
            type='submit'
            style={styles.submitButton}
            onPress={() => this.appendInput()}
        >
            <Text style={styles.submitText}>Add Contact</Text>
        </TouchableOpacity>
    </View>

    <Text style={{padding: 15, color: '#2E6CB5', fontSize: 20, paddingLeft:
'10%'}}>Contact List</Text>
    <FlatList
        data={this.state.contacts}
        renderItem={({item, index}) => this.renderContact(item, index)}
        keyExtractor={(item, index) => index.toString()}
    />
</View>

<View style={styles.container}>
    <Text style={styles.logo}>Incident Description</Text>

    <View>
        <TextInput
            name='description'
            style={styles.textInput}
            multiline={true}
            placeholder='Enter incident description here'
            onChangeText={value => this.setState({description: value})}
        />
    </View>
</View>

<View style={styles.container}>
    <View style={{flex:2}}>
        <Text style={styles.logo}>Upload Repair Quote</Text>
    </View>

    <View style={{flex:4}}>
        { loading ? (
            <ActivityIndicator />
        ) : (
            <Image source={{uri: quoteImage}} style={styles.imageContainer}
/>
        )}
    </View>

    <View style={{flex:2}}>
        <ButtonComponent
            icon='upload'
            type='antdesign'
            text='Upload from gallery'
            onPress={() => {

```

```

        this.setState({loading: true})
        this.quoteImage('cameraroll')
    }}
  />
  <ButtonComponent
    icon='camera'
    type='antdesign'
    text='Take a photo'
    onPress={() => {
      this.setState({loading: true})
      this.quoteImage('camera')
    }}
  />
</View>
</View>

<View style={styles.container}>
  <View style={{flex:2}}>
    <Text style={styles.logo}>Upload Police Report</Text>
  </View>

  <View style={{flex:4}}>
    { loading ? (
      <ActivityIndicator />
    ) : (
      <Image source={{uri: reportImage}} style={styles.imageContainer}
/>>
    )}
  </View>

  <View style={{flex:2}}>
    <ButtonComponent
      icon='upload'
      type='antdesign'
      text='Upload from gallery'
      onPress={() => {
        this.setState({loading: true})
        this.reportImage('cameraroll')
      }}
    />
    <ButtonComponent
      icon='camera'
      type='antdesign'
      text='Take a photo'
      onPress={() => {
        this.setState({loading: true})
        this.reportImage('camera')
      }}
    />
  </View>
</View>

<View style={styles.container}>
  <View style={{flex:2}}>
    <Text style={styles.logo}>Complete Claim</Text>
  </View>

  <View style={{flex:4}}>
    <View style={styles.validationItem}>

```

```

<Text>Upload at least one related contact</Text>
{ this.state.contacts.length > 0 ? (
  <Icon
    name='checksquareo'
    type='antdesign'
    color='green'
  />
) : (
  <Icon
    name='closesquareo'
    type='antdesign'
    color='red'
  />
)}
</View>
<View style={styles.validationItem}>
  <Text>Enter an incident description</Text>
  { this.state.description === null ? (
    <Icon
      name='closesquareo'
      type='antdesign'
      color='red'
    />
  ) : (
    <Icon
      name='checksquareo'
      type='antdesign'
      color='green'
    />
  )}
</View>
<View style={styles.validationItem}>
  <Text>Upload repair quote</Text>
  { this.state.quoteImage === null ? (
    <Icon
      name='closesquareo'
      type='antdesign'
      color='red'
    />
  ) : (
    <Icon
      name='checksquareo'
      type='antdesign'
      color='green'
    />
  )}
</View>
<View style={styles.validationItem}>
  <Text>Upload police report</Text>
  { this.state.reportImage === null ? (
    <Icon
      name='closesquareo'
      type='antdesign'
      color='red'
    />
  ) : (
    <Icon
      name='checksquareo'
      type='antdesign'

```



```

        color='green'
      />
    })
  </View>
  <View style={{width: '80%', alignSelf: 'center', paddingTop: '5%'}}>
    <Text style={{paddingBottom: 20}}>Enter the amount you are
requesting for this claim. Align this amount with the quote you have from a mechanic or
bodyshop.</Text>
    <FormInput
      icon=''
      type='antdesign'
      placeholder='Requested amount'
      onChangeText={value => this.setState({claimAmount: value})}
    />
  </View>
</View>

  <View style={{flex:1}}>
    <ButtonComponent
      icon='paperclip'
      type='antdesign'
      text='File Claim'
      onPress={() => {
        this.submitClaim()
      }}
    />
  </View>
</View>
</Pages>
)
}
}
const pickerStyle = {
  fieldContainer: {
    borderBottomWidth: 1,
    borderBottomColor: '#2E6CB5',
    alignSelf: 'center',
    marginBottom: 10
  },
  fieldText: {
    color: '#2E6CB5',
    fontSize: 16,
    fontWeight: '600',
    paddingHorizontal: 20,
    paddingVertical: 12,
  },
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: '10%'
  },
  imageContainer: {
    width: 300,
    height: 300,
    borderRadius: 10,
    alignSelf: 'center'

```

```

},
headerContainer: {
  flex: 3,
  flexDirection: 'row',
  alignItems: 'flex-end',
  padding: 20
},
contentContainer:{
  flex: 6,
  padding: 20,
  justifyContent: 'space-between'
},
text: {
  fontSize: 16
},
textInput:{
  borderColor: '#2E6CB5',
  borderRadius: 10,
  borderWidth: 1,
  width: '80%',
  alignSelf: 'center',
  padding: 10,
  height: '80%',
  textAlignVertical: 'top'
},
logo:{
  fontWeight:"bold",
  fontSize:35,
  color:"#2E6CB5",
  marginBottom:40,
  textAlign: 'center',
  paddingTop: '10%'
},
form: {
  borderColor: '#2E6CB5',
  borderRadius: 10,
  borderWidth: 1,
  width: '90%',
  alignSelf: 'center',
  padding: 10,
  textAlignVertical: 'top'
},
submitButton: {
  alignSelf: 'flex-end',
  paddingRight: 10,
},
submitText: {
  color: '#2E6CB5',
  borderBottomWidth: 1
},
validationItem: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  width: '70%',
  alignSelf: 'center',
  paddingBottom: 10
}
});

```

```
export default FileClaim;
```