**Chapter 3**

**SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS**

**A. Summary of Finding**

After a detailed study on the BISU Information Hub App for educational tool purposes, conducted at Bohol Island State University Balilihan Campus, Magsija, Balilihan, Bohol the researchers were able to reach on following findings.

:

1. The students cannot get latest information directly concerning to their quizzes, exam dates, assignments and organization meetings when they are not inside the campus.
2. Difficulty in communicating between teaching faculty and BISU Balilihan Campus students.
3. Time consuming in posting announcements into the bulletin board.
4. Communication difficulties of the students in their learning process, especially on how effectively communicate among themselves for group assignments, campus activities and general questions
5. Usb flash drives that have been used in sharing files may contain virus that can corrupt any other files in a laptop or computer.

# B. Conclusions

Based on the findings of the study, the researchers came up with the following conclusions:

1. Students can get latest information directly concerning to their quizzes, exam dates, assignments and organization meetings when they are not inside the campus using the app.
2. It can enhance the communication between the teaching faculty and to the students through exchanging information inside and outside the campus by the used of BISU Information Hub App.
3. Instructors can easily post announcements to the announcements tab by not consuming more effort and time.
4. It serves as an educational tool and a way of enhancing student’s learning process through communicating among themselves especially in their group assignments and campus activities.

# C. Recommendation

The researchers recommended the creation of a system, BISU Information and implement it to address the current problem. Training of the authorized personnel should also be conducted.

**Action Plan**

**BISU Information Hub App**

# Rationale

Through technology the life of human living has changed in everything it deals with. Technology is the powerful tool of innovation and invention. An advanced creation of technologies has added impression to the people of the society. One of these is the development and enhancement of new expertise. Most people of today’s generation are focused most on the practical application of knowledge, where they can relax and give themselves a better way of learning and pleasure.

People applied the modern technology in an easy manner but in an accurate evolvement of its structure and basis. Involved with this, innovation and invention come into real life. Thus, BISU Information Hub App offers learning to the user and achieved good benefits to the instructors and students. The implementation of proposed BISU Information Hub App aims to achieve these objectives:

**Objectives**

1. To enhance the communication between teaching faculty and students.
2. To provide user-friendly application that can be accessed through Android phones.
3. To implement the created android application.

# Schedule of Implementation

The proposed BISU Information Hub App will be implemented when the system is finished and if the administration of the BISU Balilihan-Campus will be convinced to use the said application.

**Mechanism of Implementation**

The BISU Information Hub App will be introduced first to the Bohol Island State University Balilihan Campus. To have knowledge on the system, the researchers will be conducting a testing on a system and will assist the user on how to operate the buttons and the basic information and instructions of the application. In order to complete the implementation, user must follow this procedure. First, install BISU Information Hub App on the phone by tapping the BISU Information Hub App.apk. Then wait for the installation process to be completed and create an account.

**REFERENCE LIST**

# A. Books

[[1]](http://www.sciencedirect.com/science/article/pii/S2212667813000063#bbib0005) Scornavacca, Huff, & Marshall (2009). Theory of Instant Messaging

[[2]](http://www.sciencedirect.com/science/article/pii/S2212667813000063#bbib0005)Basso (2008). Structuration Theory by De Sanctis: Examines how people interact and understand with the new technology

**B. Internet Reference**

http://www.jite.org/documents/Vol13/JITEv13ResearchP217-231Bouhnik0601.pdf

http://www.mastersthesiswriting.com/

1. **Unpublished Materials**

Manlangit, Fe, et.al (2010) Insurance File Management System -Unpublished Undergraduate thesis. BISU Balilihan Campus, Magsija, Balilihan, Bohol.

**APPENDICES**

**APPENDIX A**

Republic of the Philippines

BOHOL ISLAND STATE UNIVERSITY

Balilihan Campus

Magsija, Balilihan, Bohol

July 19, 2018

BENJAMIN N. OMAMALIN, Ph.D., Co.E.

Campus Dean

Bohol Island State University, Balilihan Campus

Magsija, Balilihan, Bohol

Sir:

Greetings of peace!

We, the Bachelor of Science in Information Technology senior students of Bohol Island State University – Balilihan Campus are required to propose a system as one of the requirements of our course. Our group had chosen the BISU BALILIHAN CAMPUS as our respondent.

In this regard, we would like to ask permission from your good office to allow us to conduct an interview, survey, site observation and some document browsing in order to have the data that we need for the completion and success of our study.

We are hoping for your favorable cooperation and full support for this study.

Thank you and Godspeed.

Respectfully yours,

**Kent M. Hipos  
Christian F. Bitas  
Jonmar U. Dacuba**

Noted: Approved by:

**MATHENY G. RAMO BENJAMIN N. OMAMALIN, Ph.D., Co.E.**

Research Instructor The Dean

**Appendix B**

**Modeling tools**

External Entity

* It indicates the origin of the data and it’s predetermined end.
* It describes external entities referred to as person or a group of department which the system communicates.

Process Symbol

* Symbol used for processing data into useful information.
* It describes a part of a system that transforms data or inputs into outputs in the same way.

Lines

* It shows the direction or movement of the flow of the system.

Decision

* It shows the decision making of the system.
* Symbol used for making a decision.

Terminator

* It indicates the starting and ending point of the system.

Data Store

* It indicates where the data is stored.
* It shows the collection of data.

**Appendix C**

**Interview Guide Questions**

1. How are the processes or flow in fetching information by the students in BISU Information Hub App?
2. What features can you suggest in the proposed action plan?
3. Is BISU Information Hub App easy and efficient to use?
4. What are the advantages and disadvantages of this application?
5. Do you think that the application would be helpful for the teachers and students as a way communication?

**Appendix D**

**Source Code**

**Login**

import { Actions } from 'react-native-router-flux';

import {BASE\_URL,CLIENT\_ID,CLIENT\_SECRET,GRANT\_TYPE} from '../Settings';

import axios from 'axios';

import {

LOGIN\_USER\_SUCCESS,

LOGIN\_USER\_FAIL,

LOGIN\_USER,

LOGIN\_UPDATE

} from './types';

export const loginUpdate = ({ prop, value }) =>{

return {

type: LOGIN\_UPDATE,

payload: { prop, value }

};

}

export const setUser = (user) => {

return { type: 'SET\_USER',

payload: user

}

}

export const loginUser = ({ email, password }) => {

return (dispatch) => {

let data = {

client\_id: CLIENT\_ID,

client\_secret: CLIENT\_SECRET,

grant\_type:GRANT\_TYPE,

username: email,

password: password

};

dispatch({ type: LOGIN\_USER });

const options = {

method: 'POST',

data:data,

url: BASE\_URL+'/oauth/token',

};

axios(options).then(token => {

console.log(token);

loginUserSuccess(dispatch, token.data.access\_token)

}) .catch((error) => {

console.log(error);

console.log(error.response);

loginUserFail(dispatch)});

};

};

const loginUserFail = (dispatch) => { dispatch({ type: LOGIN\_USER\_FAIL });

};const loginUserSuccess = (dispatch, token) => {

axios.defaults.headers.common['Authorization'] = 'Bearer ' + token;

axios.get('api/profile').then( result => {

dispatch({ type: 'SET\_USER',payload:result.data.data}); dispatch({ type: LOGIN\_USER\_SUCCESS});

Actions.main\_screen();

}).catch( error => {console.log('Not found!');

})

};

**Main**

import React, { Component } from 'react';

import { Container, Header, Content, Footer, FooterTab,Item,Input,Button,

Icon,ListItem,List,Left,Right,Title,Thumbnail,Body,Badge} from 'native-base';

import {

StyleSheet, View, Text, FlatList,

BackHandler

} from 'react-native';

import {Actions} from 'react-nativerouterflux';

import ConversationList from './Message/ConversationList';

import GroupList from './Group/GroupList';

import Profile from './Profile/Profile'

import FriendsList from './Friend/FriendsList'

import Report from './reports/Report'

import Announcement from './Announcement/Announcement'

import axios from 'axios';

class Main extends Component {

constructor(props) {super(props);

this.state = {

active: 'home',search: '',group\_searh: '',

people:[],groups: [],notification: {

group:0,friend:0,announcement:0

}

}

}

handleBackButton() {

return true;

}

notification(){

axios.get(`api/notifications`).then( result => {

console.log(result.data);

this.setState({notification: result.data

})

}).catch( error => {console.log(error);

console.log(error.response);

})

}componentDidMount(){

BackHandler.addEventListener('hardwareBackPress', this.handleBackButton);

if(this.props.tabs != null ){

this.setState({

active:'report'

});

}

}

previewFriend(friend){

Actions.preview\_people({friend: friend});

}

preivewGroup(group){axios.get(`api/group-check/${group.id}`).then( result => {

Actions.group\_item({group:group})

}).catch( error => {console.log(error);

console.log(error.response);

Actions.group\_preview({group:group});

})

}searchGroup(text){

this.setState({group\_searh:text

}, () => {

axios.get(`api/group-search?keyword=${this.state.group\_searh}`).then( result => console.log(result);

this.setState({groups:result.data.data

})

}).catch(error => {

console.log(error.response);

})

})

}

previewGroup(){

}renderHeader(){

if(this.state.active == 'friends'){

return ( <Header searchBar rounded >

<Item><Icon name="ios-search" />

<Input placeholder="Search Friend"

autoCorrect={false}value={this.state.search}

onChangeText={this.goSearch.bind(this)}/>

<Icon name="ios-people" /></Item>

<Button transparent>

<Text>Search</Text>

</Button>

</Header>

);

}

else if(this.state.active == 'group'){

return (<Header searchBar rounded >

<Item><Icon name="ios-search" />

<Input placeholder="Search Group"

autoCorrect={false}

value={this.state.group\_searh}

onChangeText={this.searchGroup.bind(this)}

/>

<Icon name="ios-people" />

</Item><Button transparent>

<Text>Search</Text>

</Button></Header>

) }

else if(this.state.active == 'announcement'){

return (

<Header>

<Left/>

<Body>

<Title>Announcement</Title>

</Body><Right /></Header>

) }

else if(this.state.active == 'home'){

return ( <Header><Left/><Body>

<Title>Messages</Title></Body>

<Right /> </Header>

) }

else if(this.state.active == 'report'){

return (<Header><Left/><Body>

<Title>Reports</Title>

</Body><Right /></Header>

) }

else{ return <Header/>

}

}

renderView(){

if(this.state.group\_searh.length > 0){

return (<Content><FlatList

data={this.state.groups}

keyExtractor={(item,Index)=> `group\_${item.id}`}

renderItem={({item,index}) =>

<List><ListItem avatar button={true} onPress={this.preivewGroup.bind(this,item)}<Left><Thumbnailsource={require('../assets/images/da.png')} /></Left><Body>

<Text>{item.name}</Text>

<Text note>Creator</Text></Body><Right>

</Right></ListItem></List>

}/></Content>

)}

if(this.state.search.length > 0){

return (<Content><FlatList

data={this.state.people}

keyExtractor={(item, index) => `pending\_${item.id}`}

renderItem={({item,index}) =>

<List><ListItem avatar button={true} onPress={this.previewFriend.bind(this,item)}> <Left><Thumbnail source={{ uri: item.person.avatar }} />

</Left><Body><Text>{item.name}</Text>

<Text note>{item.person.course}</Text>

</Body><Right></Right></ListItem></List>

}/></Content>

) }

if(this.state.active == 'home'){

return <ConversationList />;

}

else if(this.state.active == 'group'){

return <GroupList />;

}else if(this.state.active == 'profile'){

return <Profile />

}else if(this.state.active == 'friends'){

return <FriendsList />

}

else if(this.state.active == 'announcement'){

return <Announcement />

}

else if(this.state.active == 'report'){

return <Report/>

}

else {

return (

<View><Text> Under construction </Text>

</View>

) }

}

changeView(tab){

this.notification()

if(tab == 'friends'){

axios.get(`api/read-notification?type=FriendRequestNotification`)}

else if(tab == 'announcement'){

axios.get(`api/read-notification?type=AnnouncementNotification`)}

else if(tab == 'group'){

axios.get(`api/read-notification?type=GroupNotification`)

}

this.setState({active: tab, search: '', group\_searh: ''})

}

goSearch(text){ console.log(text);

this.setState({search:text}, () => {

axios.get(`api/get-people?keyword=${this.state.search}`).then( result => {console.log(result);

this.setState({

people:result.data.data

})

}).catch(error => {

console.log(error.response);

})

}) }

renderBadge(number){

if(number > 0){

return (<Badge><Text style={{ fontSize: 10}}>{number}</Text></Badge>

) }return null;

}

render() {return (

<Container>{this.renderHeader()}

<Content>{this.renderView()}

</Content><Footer><FooterTab>

<Button active onPress={

this.changeView.bind(this,'home')}>

<Icon name="home" /></Button>

<Button onPress={this.changeView.bind(this,'group')} badge={this.state.notification.group > 0} vertical>

{this.renderBadge(this.state.notification.group)} <Icon name="md-contacts" />

</Button><Button onPress={this.changeView.bind(this,'profile')}> <Icon name="md-contact" /></Button>

<Button onPress={this.changeView.bind(this,'friends')} badge={this.state.notification.friend > 0} vertical>

{this.renderBadge(this.state.notification.friend)} <Icon name="people" /></Button>

<Button onPress={this.changeView.bind(this,'announcement')} badge={this.state.notification.announcement > 0} vertical>

{this.renderBadge(this.state.notification.announcement)}

<Icon name="md-notifications" /></Button>

<Button onPress={this.changeView.bind(this,'report')}> <Icon type="FontAwesome" name="file" />

</Button>

</FooterTab></Footer></Container>

);

}

}

const styles = StyleSheet.create({

});

**Message**

'use strict';

import React, { Component } from 'react';

import {StyleSheet, View,Text, Platform,

Clipboard,

Linking

}

from 'react-native';

import {GiftedChat, Actions, Bubble} from 'react-native-gifted-chat';

import CustomActions from './CustomActions';

import CustomView from './CustomView';

import axios from 'axios';

import { connect } from 'react-redux';

import {BASE\_URL,SOCKET\_URL} from '../../Settings';

import SocketIOClient from 'socket.io-client';

import {ActionSheet} from 'native-base';

class Chat extends Component {

constructor(props) {super(props);

this.socket=SocketIOClient(SOCKET\_URL);

this.state = {messages: [], loadEarlier: true,

typingText: null, isLoadingEarlier: false,

};

this.\_isMounted = false;

this.onSend = this.onSend.bind(this);

this.onReceive = this.onReceive.bind(this);

this.renderCustomActions = this.renderCustomActions.bind(this);

this.renderBubble = this.renderBubble.bind(this); this.renderFooter = this.renderFooter.bind(this);

this.onLoadEarlier = this.onLoadEarlier.bind(this);

this.\_isAlright = null;

}

componentDidMount(){

axios.get(`api/conversation/${this.props.conversation.id}`).then( result => {

this.setState({

messages: result.data.data.reverse()

})

}).catch( error => {

console.log(error.response)

})this.socket.emit('join',`chat\_${this.props.conversation.id}`);

this.socket.on('message', (data) => {

if(data != ''){

const { messages} = this.state;

messages.unshift(data)

this.setState({

messages: messages.splice(0)

});

}

});

}

componentWillUnmount() {

this.\_isMounted = false;

}

onLoadEarlier() {

this.setState((previousState) => {

return {isLoadingEarlier: true,

};

});

setTimeout(() => {

if (this.\_isMounted === true) {

this.setState((previousState) => {

return{messages: GiftedChat.prepend(previousState.messages,oldMsg), loadEarlier: false,

isLoadingEarlier: false,

};

});

}

10); // simulating network

}onDelete(messageIdToDelete) {

this.setState(previousState =>

({messages: previousState.messages.filter(message => message.id !== messageIdToDelete) }))

}

onLongPress(contextcontext,messasge) {

console.log(this.state.messages);

console.log(messasge.\_id);

let label = 'Copy Text';

let isImage = messasge.type == 'image';

if(isImage){

label = 'Download Image'

}

const options = [ label, 'Delete', 'Cancel',];

const cancelButtonIndex = options.length - 1; ActionSheet.show({

options: options,

cancelButtonIndex: cancelButtonIndex,

}, (buttonIndex) => {

switch (buttonIndex) {

case 0:

if(isImage){

let url = messasge.text;

Linking.canOpenURL(url).then(supported => {

if (supported) {

Linking.openURL(url);

} else {

console.log("Don't know how to open URI: " + url); }

});

}else{

Clipboard.setString(messasge.text);

}

break

case 1:

axios.post(`api/delete-message/${messasge.\_id}`).then( result => console.log(result))

.catch( error => {

console.log(error);

console.log(error.response);

})

this.setState(previousState =>

({messages: previousState.messages.filter(msg => msg.\_id !== messasge.\_id) }))

break;

}

});

console.log(messasge);

}

onSend(messages= []) {

console.log(messages);

const { conversation } = this.props;

messages.map( message => {

if(message.hasOwnProperty('image')){

let data = new FormData();

data.append('image', {uri: message.image,

type: 'image/jpeg', name: 'testPhotoName'

});data.append('message', '');

data.append('conversation\_id',conversation.id);

axios.post(`api/message`,data, {headers: {

'Content-Type': 'multipart/form-data' }

}).then( result =>{console.log(result);

this.socket.emit('message',result.data.data)

}).catch( error =>{

console.log(error.response);

})

}

if(message.hasOwnProperty('text')){

console.log(message.text);

let data ={message: message.text,

conversation\_id: conversation.id }

axios.post(`api/message`,data).then( result =>{console.log(result.data.data);

this.socket.emit('message',result.data.data)

}).catch( error =>{

console.log(error.response);

})

}

})

}

answerDemo(messages) {

if (messages.length > 0) {

if((messages[0].image|| messages[0].location) || !this.\_isAlright) {

this.setState((previousState) => {

return {typingText: 'React Native is typing'

};

});

}

}

setTimeout(() => {

if (this.\_isMounted === true) {

if (messages.length > 0) {

if (messages[0].image) {

this.onReceive('Nice picture!');

} else if (messages[0].location) {

this.onReceive('My favorite place');

} else {if (!this.\_isAlright) {

this.\_isAlright = true;

this.onReceive('Alright');

}

}

}

}

this.setState((previousState) => {

return {typingText: null, };});

}, 1000); }

onReceive(text) {

this.setState((previousState) => {

return {

messages: GiftedChat.append(previousState.messages, {

id: Math.round(Math.random() \* 1000000),

text: text, createdAt: new Date(),

user: {\_id: 2, name: 'React Native',

//avatar: 'https://facebook.github.io/react/img/logo\_og.png', },

}),

};

});

}

renderCustomActions(props) {

return (

<CustomActions

{...props}/>);

if (Platform.OS === 'ios') {

}const options = {'File': (props) => {

alert('option 1'); },'Photo': (props) => {

alert('option 2'); },'Cancel': () => {},

};return (<Actions{...props}options={options}

/>

);

}

renderBubble(props) {

return (<Bubble{...props}wrapperStyle={{

left: {backgroundColor: '#f0f0f0', }

}}/>

);

}

renderCustomView(props) {

return (<CustomView{...props}/>

);

}

renderFooter(props) {

if (this.state.typingText) {

return (

<View style={styles.footerContainer}>

<Text style={styles.footerText}>

{this.state.typingText}</Text></View>

);

}

return null; }

render() {

return (<GiftedChat

messages={this.state.messages}

onSend={this.onSend}

onDelete={this.onDelete}

onLongPress={this.onLongPress.bind(this)}

user={{\_id: this.props.user.id, // sent messages should have same user.\_id

}}

renderActions={this.renderCustomActions}

renderBubble={this.renderBubble}

renderCustomView={this.renderCustomView}

renderFooter={this.renderFooter}/> );

}

}

const styles = StyleSheet.create({

footerContainer: { marginTop: 5,

marginLeft: 10, marginRight: 10,

marginBottom: 10, },footerText: {fontSize: 14, color: '#aaa',

},

});

const mapStateToProps = ({ auth }) => {

const { user} = auth;

return { user };

};

export default connect(mapStateToProps, {})(Chat);

**Group Request**

import React, { Component } from 'react'

import { View, StyleSheet, FlatList,ScrollView } from 'react-native'

import { Container, List, ListItem, Thumbnail, Left, Body, Right, Text, Button, Separator } from 'native-base'

import axios from 'axios';

export class GroupRequest extends Component {

constructor(props) {super(props);

this.state = {data: [],loading: false, }

}componentDidMount(){

axios.get(`api/group-requests/${this.props.group.id}`).then( result => {console.log(result);

this.setState({ data: result.data.data })

}).catch( error => {console.log(error)

console.log(error.response)

})

}

accept(friend,index){

this.setState({

loading:true

})

let {data} = this.state

axios.post(`api/group-accept-request/${friend.group.id}/${friend.user.id}`).then( result =>{

data.splice(index,1); setTimeout( () => {

this.setState({ data:data.splice(0),

loading:false

})

} ,3000)

}).catch( error => { console.log( error) })

}

decline(friend,index){

let {data} = this.state

this.setState({ loading:true

})

axios.post(`api/group-decline-request/${friend.group.id}/${friend.user.id}`).

then( result =>{ data.splice(index,1);

this.setState({ data: data.splice(0),

loading:false

}) }).catch( error => {

console.log( error.response)

})

}render() {

return (

<Container style={styles.wrapper}>

<ScrollView><FlatList

data={this.state.data}

keyExtractor={(item, index) => `pending\_${item.id}`}

renderItem={({item,index}) =>

<List><ListItem thumbnail> <Left>

<Thumbnail square

source={require('../../assets/images/da.png')} />

</Left><Body>

<Text>{item.user.name}</Text>

<Textnote numberOfLines={1}>{item.type}</Text>

</Body><Right><Right>

<View style={{flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center'}}>

<Button success rounded onPress={this.accept.bind(this,item,index)} disabled={this.state.loading}><Text style={{ fontSize: 10}}> Accept </Text></Button>

<Button danger rounded onPress={this.decline.bind(this,item,index)} disabled={this.state.loading}><Text style={{ fontSize: 10}}> Decline </Text></Button>

</View></Right></Right></ListItem>

</List>

}

/>

</ScrollView>

</Container>

)}

}

const styles = StyleSheet.create({

})

export default GroupRequest

**Group Chats**

'use strict';

import React, { Component } from 'react';

import {

StyleSheet, View, Text, Platform

} from 'react-native';

import {GiftedChat, Actions, Bubble} from 'react-native-gifted-chat';

import CustomActions from '../Message/CustomActions';

import CustomView from '../Message/CustomView';

import {BASE\_URL,SOCKET\_URL} from '../../Settings';

import SocketIOClient from 'socket.io-client';

import axios from 'axios';

const oldMsg = [{

\_id: Math.round(Math.random() \* 1000000),

createdAt: new Date(Date.UTC(2016, 7, 30, 17, 20, 0)),

user: {\_id: 1,

name: 'Developer', },},{

id: Math.round(Math.random() \* 1000000),

createdAt: new Date(Date.UTC(2016, 7, 30, 17, 20, 0)),

user: {

\_id: 2,

name: 'Mike',

},

},

{\_id: Math.round(Math.random() \* 1000000),

createdAt: new Date(Date.UTC(2016, 7, 30, 17, 20, 0)),

user: {\_id: 1, name: 'Developer',

},},];

class GroupChat extends Component {

constructor(props) {super(props);

this.socket = SocketIOClient(SOCKET\_URL);

this.state = {messages: [],loadEarlier: true,

typingText: null, isLoadingEarlier: false,

profile: {}};

this.\_isMounted = false;

this.onSend = this.onSend.bind(this);

this.renderCustomActions = this.renderCustomActions.bind(this);this.renderBubble = this.renderBubble.bind(this);

this.renderFooter = this.renderFooter.bind(this);

this.onLoadEarlier = this.onLoadEarlier.bind(this);

this.\_isAlright = null; }

componentDidMount(){

axios.get('api/profile').then( result => {

console.log(result);

this.setState({profile: result.data.data })

}).catch( error => {

})

axios.get(`api/conversation/${this.props.group.conversation\_id}`).then( result => {

this.setState({

messages: result.data.data.reverse()

})

}).catch( error => {

console.log(error.response)

})

this.socket.emit('join',

`chat\_${this.props.group.conversation\_id}`);

this.socket.on('message', (data) => {

if(data != ''){ const { messages} = this.state;

messages.unshift(data)

this.setState({

messages: messages.splice(0) });

}

});

}

componentWillUnmount() {

this.\_isMounted = false; }

onLoadEarlier() {

this.setState((previousState) => {

return {

isLoadingEarlier: true,

};

});

setTimeout(() => {

if (this.\_isMounted === true) {

this.setState((previousState) => {

return

messages: GiftedChat.prepend(previousState.messages,oldMsg),

loadEarlier: false,isLoadingEarlier: false,};

});

}

}, 10);

}

onSend(messages= []) {

const {group} = this.props

messages.map( message => {

if(message.hasOwnProperty('image')){

let data = new FormData();

data.append('image', {uri: message.image,

type: 'image/jpeg', name: 'testPhotoName'

});data.append('message', '');

data.append('conversation\_id',group.conversation\_id);

axios.post(`api/message`,data, {

headers: {

'Content-Type': 'multipart/form-data'

}

}).then( result =>{console.log(result);

this.socket.emit('message',result.data.data)

}).catch( error =>{

console.log(error.response);

})

}

if(message.hasOwnProperty('text')){

let data ={

message: message.text,

conversation\_id: group.conversation\_id

} axios.post(`api/message`,data).then( result =>{ console.log(result.data.data);

this.socket.emit('message',result.data.data)

}).catch( error =>{

console.log(error.response);

})

}

})

}

renderCustomActions(props) {

return (<CustomActions{...props}/>

);

}

renderBubble(props) {

return (<Bubble {...props} wrapperStyle={{

left: { backgroundColor: '#f0f0f0', } }} />

);

}renderCustomView(props) {

return (

<CustomView {...props} />

);

}

renderFooter(props) {

if (this.state.typingText) {

return (

<View style={styles.footerContainer}>

<Text style={styles.footerText}>

{this.state.typingText} </Text></View>

);

}

return null;

render() {

return (

<GiftedChat

messages={this.state.messages}

onSend={this.onSend}

user={{\_id: this.state.profile.id, // sent messages should have same user.\_id

}}renderActions={this.renderCustomActions}

renderBubble={this.renderBubble}

renderCustomView={this.renderCustomView}

renderFooter={this.renderFooter}/>

);

}

}

const styles = StyleSheet.create({

footerContainer: { marginTop: 5,

marginLeft: 10, marginRight: 10,

marginBottom: 10,

},footerText: {

fontSize: 14,color: '#aaa',

},

});

export default GroupChat;

**Group Announcements**

import React, { Component } from 'react'

import {

View, StyleSheet, FlatList,Modal,ScrollView } from 'react-native'

import { DatePicker,Container, List, ListItem, Thumbnail, Left, Body, Right, Text, Button, Separator,Form,Label,Item,Input,Content,H1 } from 'native-base'

import axios from 'axios';

import {Actions} from 'react-native-router-flux';

import moment from 'moment';

export class GroupAnnouncement extends Component {

constructor(props) {super(props);

this.state = { title: '', body: '', modalVisible: false,

data\_p: [],

data\_n: [],

isTeacher: false,

chosenDate: new Date(),

loading:false

}

this.setDate = this.setDate.bind(this);

}

setDate(newDate) {

this.setState({ chosenDate: newDate });

}

componentDidMount(){

axios.get('api/profile').then( result => {

if(result.data.data.type == 'Teacher'){

this.setState({ isTeacher: true })

}

}).catch( error => {

})

axios.get(`api/group-announcement/${this.props.group.id}`).then( result => {

console.log(result.data)

this.setState({

data\_p: result.data.priority,

data\_n:result.data.normal

})

}).catch( error => {

console.log(error)

console.log(error.response)

})

}

create(){

this.setState({ modalVisible: true

})

}

save(){

this.setState({ loading:true })

const {title,body,data\_p, date\_ended} = this.state;

let ended = moment(date\_ended).format('YYYY-MM-DD');

axios.post(`api/group-announcement/${this.props.group.id}`, {title,body,date\_ended:ended}).then( result => {

data\_p.unshift(result.data.data);

this.setState({

modalVisible: false, data\_p:data\_p.splice(0),

title: '',body: '', date\_ended: new Date(),

loading:false })

}).catch( error => {console.log(error)

console.log(error.response)

})

}renderButton(){

if(this.state.isTeacher)

return (

<Button success full rounded style={styles.buttonStyle} onPress={this.create.bind(this)} disabled={this.props.loading}>

<Text> Create Announcement </Text>

</Button>

)

return null;

}

previewContent(item){

Actions.preview\_announcement({

content:item

});

}

render() {

return (

<Container style={styles.wrapper}>

<ScrollView> <Separator>

<Text>Priority</Text></Separator>

<FlatList data={this.state.data\_p}

keyExtractor={(item, index) => `a\_${item.id}`} renderItem={({item}) =>

<List><ListItem thumbnail><Left>

<Thumbnailsquaresource={require('../../assets/images/da.png')} />

</Left><Body><Text>{item.title}</Text>

<Text note numberOfLines={1}>{item.created\_at}

</Text></Body><Right><Button transparent onPress={this.previewContent.bind(this,item)}>

<Text>View</Text></Button></Right>

</ListItem></List> } /><Separator>

<Text>Normal</Text></Separator>

<FlatList data={this.state.data\_n}

keyExtractor={(item, index) => `b\_${item.id}`}

renderItem={({item}) =>

<List><ListItem thumbnail><Left>

<Thumbnail square source={require('../../assets/images/da.png')} />

</Left><Body><Text>{item.title}</Text>

<Textnote numberOfLines={1}>{item.created\_at}

</Text></Body><Right><Button transparent onPress={this.previewContent.bind(this,item)}> <Text>View</Text>

</Button></Right></ListItem>

</List> }/></ScrollView>

<Modal

animationType="slide"

transparent={false}

visible={this.state.modalVisible}

onRequestClose={() => {

Alert.alert('Modal has been closed.');

}}>

<View style={{marginTop: 22}}>

<View style={{ alignItems: 'center'}}>

<H1>Create Announcement</H1>

</View><Form>

<Label>Date Ended</Label>

<DatePicker defaultDate={new Date()}

minimumDate={new Date()} locale={"en"}

modalTransparent={false}

animationType={"fade"}

androidMode={"spinner"}

placeHolderText="Select date"

textStyle={{ color: "green" }}

placeHolderTextStyle={{ color: "#d3d3d3" }}

onDateChange={this.setDate} />

<Item floatingLabel><Label>Title</Label>

<Input onChangeText={(text) =>

this.setState({title:text})}

value={this.state.title} /> </Item>

<Item floatingLabel last>

<Label>Body</Label>

<Input multiline={true}numberOfLines={4}

onChangeText={(text) => this.setState({body:text})}

value={this.state.body}/> </Item>

</Form><Button success full rounded style={styles.buttonStyle}onPress={this.save.bind(this)} disabled={this.props.loading}>

<Text>Save </Text></Button></View>

</Modal> {this.renderButton()}

</Container> )}

}

const styles = StyleSheet.create({

})

export default GroupAnnouncement

**Group Members**

import React, { Component } from 'react'

import { View, StyleSheet, FlatList,ScrollView } from 'react-native'

import { Container, List, ListItem, Thumbnail, Left, Body, Right, Text, Button, Separator } from 'native-base'

import axios from 'axios';

import {Actions} from 'react-native-router-flux';

export class GroupMember extends Component {

constructor(props) { super(props);

this.state = {members: [], isTeacher: false,

profile:{}

}

}

componentDidMount(){

axios.get('api/profile').then( result => {

this.setState({ profile: result.data.data

})

if(result.data.data.type == 'Teacher'){

this.setState({ isTeacher: true

})

}

}).catch( error => {

})

axios.get(`api/group-members/${this.props.group.id}`).then( result => {console.log(result.data.data);

this.setState({ members: result.data.data

})

}).catch( error => { console.log(error)

console.log(error.response) })

}

remove(user,index){

axios.post(`api/remove-group/${this.props.group.id}/${user.id}`).then( result => {

let {members} = this.state

members.splice(index,1);

this.setState({

members: members.splice(0)

})

}).catch( error => { console.log(error)

console.log(error.response)

})

}

renderButton(item,index){

if(this.state.isTeacher){

if(this.state.profile.id != item.id)

{

return (

<Button danger rounded onPress={this.remove.bind(this,item,index)}><Text style={{ fontSize: 10}}> Kick </Text></Button>

) }

} return null;

}

renderAddMember(){

if(this.state.isTeacher){

return (

<Button success full rounded onPress={this.addMember.bind(this)} disabled={this.props.loading}>

<Text> Add Member </Text>

</Button>

) }

return null;

}

addMember() { const {group} = this.props;

Actions.searchMember({group:group});

}

render() {

return (

<Container style={styles.wrapper}>

<ScrollView> <FlatList

data={this.state.members}

keyExtractor={(item, index) => `member\_${item.id}`}

renderItem={({item,index}) =>

<List> <ListItem avatar> <Left>

<Thumbnail source={{

uri: item.person.avatar }} /></Left>

<Body><Text>{item.name}</Text>

<Text note>{item.person.remark}</Text>

</Body> <Right>

{this.renderButton(item,index)} </Right>

</ListItem></List> } /> </ScrollView>

{this.renderAddMember()} </Container>

) }

}const styles = StyleSheet.create({

})

export default GroupMember

**Friends**

import React, { Component } from 'react'

import { View, StyleSheet, FlatList } from 'react-native'

import { Container, Item, Icon, Input, List, ListItem, Left, Body, Right, Thumbnail, Text, Separator, Content, Button } from 'native-base'

import axios from 'axios';

import {Actions} from 'react-native-router-flux';

export class FriendsList extends Component {

constructor(props) { super(props);

this.state = { pending: [], friends: [],

loading:false,

}

}

componentDidMount(){

axios.get('api/pending-friends').then(

result => { console.log( result.data.data);

this.setState({ pending: result.data.data

});

}).catch( error => {console.log(error);

console.log(error.response);

})

axios.get('api/friends').then( result => {

console.log( result.data.data);

this.setState({ friends: result.data.data

}, () => {

console.log(this.state.friends);

});

}).catch( error => {

console.log(error);

console.log(error.response);

})

}

accept(friend,index){

this.preventDefault = true

this.setState({ loading:true

})

let {friends,pending} = this.state

axios.post(`api/friend-accept/${friend.sender.id}`).then( result =>{

friends.unshift(friend);

pending.splice(index,1);

console.log(pending);

setTimeout( () => {

this.setState({ friends:friends.splice(0),

pending: pending.splice(0), loading:false

})

},3000 )

this.preventDefault = false;

}).catch( error => {

console.log( error.response)

this.preventDefault = false;

})

}

decline(friend,index){

this.setState({

loading:true

})

setTimeout( () => {

let {friends,pending} = this.state

pending.splice(index,1);

this.setState({

pending: pending.splice(0),

loading:false

})

},4000)

}

previewFriend(friend){

Actions.preview\_people({friend: friend});

}

renderButton(item,index){

console.log(this.state.loading);

return (

<View style={{flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center'}}>

<Button success rounded onPress={this.accept.bind(this,item)} disabled={this.state.loading}><Text style={{ fontSize: 10}} > Accept </Text></Button>

<Button danger rounded onPress={this.decline.bind(this,item,index)} disabled={this.state.loading}><Text style={{ fontSize: 10}}> Decline </Text></Button>

</View>

)}

render() {

return (

<Container style={styles.wrapper}>

<Content> <Separator bordered>

<Text>Pending List</Text> </Separator>

<FlatList data={this.state.pending}

keyExtractor={(item, index) => `pending\_${item.id}`}

renderItem={({item,index}) =>

<List onPress={this.previewFriend.bind(this,item)}>

<ListItem avatar button={true} onPress={this.previewFriend.bind(this,item)}> <Left>

<Thumbnail source={{ uri: item.person.avatar}} /> </Left>

<Body> <Text>{item.sender.name}</Text>

<Text note>{item.sender.person.remark}</Text>

</Body><Right> {this.renderButton(item,index)} </Right>

</ListItem> </List> /><Separator bordered>

<Text>Friends List</Text> </Separator>

<FlatList

data={this.state.friends}

keyExtractor={(item, index) => `friend\_${item.id}`}

renderItem={({item,index}) =>

<List>

<ListItem avatar button={true} onPress={this.previewFriend.bind(this,item)}><Left>

<Thumbnail source={{ uri: item.person.avatar }} />

</Left> <Body>

<Text>{item.sender.name}</Text>

<Text note>{item.person.remark}</Text>

</Body> <Right> </Right> </ListItem>

</List>

} /> </Content> </Container>

)

}

}

const styles = StyleSheet.create({ wrapper: {

flex: 1

},searchWrapper: {

padding: 5

}

})

export default FriendsList

**Users**

import React, { Component } from 'react';

import { Text, View, StyleSheet, Image } from 'react-native';

import { Container, Thumbnail, H3,Button } from 'native-base';

import { connect } from 'react-redux';

import {Actions} from 'react-native-router-flux';

import axios from 'axios';

class People extends Component {

constructor(props) { super(props);

this.state = { loading:false, people: {

person:{ avatar: ''

}

}

}

}

addFriend(){

this.setState({ loading:true

})

let person = this.state.people;

axios.post(`api/friend-request/${person.id}`).then( result =>{

person.status = 2;

this.setState({ people:person, loading:false

})

}).catch( error => {

console.log(error.response);

})

}

componentDidMount(){

const {friend} = this.props;

console.log(this.props.friend.person);

this.setState({ people:friend}, () =>

{ console.log(this.state.people.person.avatar)

} );

}

accept(){

let person = this.state.people;

this.setState({ loading:true

})

axios.post(`api/friend-accept/${person.id}`).then( result =>{

person.status = 3; this.setState({

people:person, loading:false

})

}).catch( error => {

console.log( error.response)

})

}

decline(){

this.setState({ loading:true

})

let person = this.state.people;

axios.post(`api/friend-deny/${person.id}`).then( result =>{

person.status = 0;

this.setState({ people:person,

loading:false

})

}).catch( error => {

console.log( error.response)

})

}

cancel(){

let person = this.state.people;

this.setState({ loading:true

})

axios.post(`api/friend-cancel/${person.id}`).then( result =>{

person.status = 0;

this.setState({ people:person,

loading:false

})

}).catch( error => {

console.log( error.response)

})

}

unfriend(){

let person = this.state.people;

const {friend} = this.props;

this.setState({ loading:true

}) console.log(friend);

axios.post(`api/unfriend/${friend.id}`).then( result =>{ person.status = 0;

this.setState({ people:person, loading:false

})

}).catch( error => {

console.log( error.response)

})

}

renderButton(){

if(this.state.people.status == 0){

return (<Button full rounded success

onPress={this.addFriend.bind(this)}

disabled={this.state.loading}>

<Text>Add Friend</Text> </Button>)

}

else if(this.state.people.status == 2){

return (<Button full danger rounded success

onPress={this.cancel.bind(this)}

disabled={this.state.loading}>

<Text>Cancel Friend Request</Text>

</Button>)

}

else if(this.state.people.status == 1){

return (

<View style={{flex: 1,flexDirection: 'row', justifyContent: 'space-evenly', alignItems: 'center'}}>

<Button success full rounded onPress={this.accept.bind(this)} disabled={this.state.loading} ><Text> Accept </Text></Button>

<Button danger full rounded onPress={this.decline.bind(this)} disabled={this.state.loading}>

<Text> Decline </Text></Button> </View>

)

}

else {

return (<Button full danger rounded

disabled={this.state.loading}

onPress={this.unfriend.bind(this)} >

<Text>Unfriend</Text> </Button>);

}

}

render() {

return (

<Container style={styles.wrapper}>

<View style={styles.profileWrapper}>

<Imagesource={require('../../assets/images/da.png')} style={styles.backgroundImage} />

<Thumbnail large style={styles.profileImage} source={{

uri: this.state.people.person.avatar}} />

<H3 style={styles.name}>{this.props.friend.person.name}</H3>

</View>

<View style={styles.infoWrapper}>

<View style={styles.idWrapper}>

<Text style={styles.label}>Sudent ID number</Text><Text style={styles.value}>

{this.props.friend.person.student\_id}</Text>

</View>

<View style={styles.detailsWrapper}>

<H3>Personal Details</H3>

<Text style={styles.infoLabel}>Course

</Text>

<Text style={styles.infoValue}>{this.props.friend.person.course}</Text>

<Text style={styles.infoLabel}>Birthdate

</Text>

<Text style={styles.infoValue}>{this.props.friend.person.birthdate} </Text>

</View> {this.renderButton()} </View>

</Container>

);

}

}

const styles = StyleSheet.create({

wrapper: { flex: 1 },

profileWrapper: { flex: 1, overflow: 'hidden',

alignItems: 'center', justifyContent: 'center',

backgroundColor: '#2196F3'

},

backgroundImage: {

width: '110%', height: '110%', opacity: 0.1,

resizeMode: 'cover', position: 'absolute'

}, profileImage: {

width: 130, height: 130, borderRadius: 75

},

name: {

marginTop: 10, color: '#ffffff'

},

infoWrapper: {

flex: 2,

padding: 5 },

idWrapper: {

alignItems: 'center', padding: 5,

paddingBottom: 10, borderBottomWidth: 1,

borderBottomColor: '#e0e0e0'

},

label: {

fontSize: 12, color: '#9e9e9e'

},

value: {

fontSize: 18, fontWeight: 'bold',

color: '#212121'

},

detailsWrapper: {

padding: 5

},infoLabel: {

fontSize: 14, marginTop: 5

},infoValue: {

fontSize: 16, color: '#212121'

}

})

export default People;

**Profile**

import React, { Component } from 'react'

import { Text, View, StyleSheet, Image } from 'react-native'

import { Container, Thumbnail, H3 ,Button} from 'native-base'

import axios from 'axios';

import {Actions} from 'react-native-router-flux';

export class Profile extends Component {

constructor(props) {

super(props);

this.state = {

profile: {

person:{ avatar: ''}

},

}

}

componentDidMount(){

axios.get('api/profile').then( result => {

console.log(result); this.setState({

profile: result.data.data })

}).catch( error => {

})

}

logout(){

Actions.login();

}

render() {

return (

<Container style={styles.wrapper}>

<View style={styles.profileWrapper}>

<Image source={require('../../assets/images/da.png')} style={styles.backgroundImage} />

<Thumbnail large style={styles.profileImage} source={{

uri: this.state.profile.person.avatar}} />

<H3 style={styles.name}>My Profile</H3>

</View>

<View style={styles.infoWrapper}>

<View style={styles.idWrapper}>

<Textstyle={styles.label}>{`${this.state.profile.type} ID number`} </Text>

<Text style={styles.value}>{`${this.state.profile.person.student\_id}`}</Text>

</View>

<View style={styles.detailsWrapper}>

<H3>Personal Details</H3>

<Text style={styles.infoLabel}>{`${this.state.profile.person.course == ''? '':'Course'}`}</Text>

<Text style={styles.infoValue}>{`${this.state.profile.person.course}`}</Text>

<Text style={styles.infoLabel}>Name</Text>

<Text style={styles.infoValue}>{`${this.state.profile.name}`}</Text>

<Text style={styles.infoLabel}>Birthdate</Text>

<Text style={styles.infoValue}>{`${this.state.profile.person.birthdate}`}</Text> </View>

<Button danger full rounded style={styles.buttonStyle} onPress={this.logout.bind(this)} disabled={this.props.loading}>

<Text>Logout </Text>

</Button> </View> </Container>

)

}

}const styles = StyleSheet.create({

wrapper: { flex: 1 },

profileWrapper: {

flex: 1, overflow: 'hidden',

alignItems: 'center' justifyContent: 'center',

backgroundColor: '#2196F3'

},backgroundImage: {

width: '110%', height: '110%', opacity: 0.1,

resizeMode: 'cover', position: 'absolute'

},profileImage: { width: 130, height: 130,

borderRadius: 75

}

name: { marginTop: 10, color: '#ffffff'

}, infoWrapper: { flex: 2, padding: 5

},

idWrapper: { alignItems: 'center', padding: 5,

paddingBottom: 10, borderBottomWidth: 1,

borderBottomColor: '#e0e0e0' },

label: { fontSize: 12, color: '#9e9e9e'

},

value: {

fontSize: 18, fontWeight: 'bold',

color: '#212121'

},detailsWrapper: { padding: 5

},

infoLabel: { fontSize: 14, marginTop: 5

},

infoValue: { fontSize: 16, color: '#212121'

}

})

export default Profile

**Reports**

'use strict';

import React, { Component } from 'react';

import {

StyleSheet, View, Modal, FlatList,

PermissionsAndroid

} from 'react-native';

import CameraRollPicker from 'react-native-camera-roll-picker';

import NavBar, { NavButton, NavButtonText, NavTitle } from 'react-native-nav';

import axios from 'axios';

import {Actions} from 'react-native-router-flux';

import { Card, CardItem , Body, Thumbnail ,Text,Container, Header, Content, Textarea, Form, Input , Item, Button } from "native-base";

class Report extends Component {

constructor(props) { super(props);

this.\_images = [];

this.state = { modalVisible: false, title: '',

'description': '', loading: false

};

this.selectImages = this.selectImages.bind(this);

}

componentDidMount(){

this.requestCameraPermission();

}

async requestCameraPermission() {

try {

const granted = await PermissionsAndroid.request(

PermissionsAndroid.PERMISSIONS.READ\_EXTERNAL\_STORAGE,

{

'title': 'Cool Photo App Camera Permission',

'message': 'Cool Photo App needs access to your camera ' + 'so you can take awesome pictures.'

})

if (granted === PermissionsAndroid.RESULTS.GRANTED) {

console.log("You can use the camera")

} else {

console.log("Camera permission denied")

}

} catch (err) { console.warn(err)

} }

setImages(images) { this.\_images = images;

}

clearImages(){

Actions.main\_screen({tabs: 'reports'});

}

getImages() {

return this.\_images;

}

setModalVisible(visible = false) {

this.setState({modalVisible: visible});

}

submitReport(){

this.setState({ loading:true

})

let data = new FormData();

this.getImages().map( image => {

data.append('images[]', {

uri: image.uri,

type: 'image/jpeg',

name: 'testPhotoName'

});

})

data.append('title', this.state.title);

data.append('description',this.state.description);

axios.post(`api/report`,data, {

headers: {

'Content-Type': 'multipart/form-data'

}

}).then( result =>{

this.setState({

title:'', description: '', loading: false

});

this.setImages([]);

this.clearImages();

alert('Report Sent');

}).catch( error =>{

alert('Need more supporting evidence');

console.log(error.response);

})

}

renderNavBar() {

return (

<NavBar style={{ statusBar: {

backgroundColor: '#FFF',},

navBar: { backgroundColor: '#FFF',},

}}>

<NavButton onPress={() => {

this.setModalVisible(false);}}>

<NavButtonText style={{color: '#000',}}>

{'Cancel'} </NavButtonText> </NavButton>

<NavTitle style={{ color: '#000', }}>

{'Camera Roll'} </NavTitle>

<NavButton onPress={() => {

this.setModalVisible(false); }}>

<NavButtonText style={{ color: '#000', }}>

{'Ok'} </NavButtonText> </NavButton>

</NavBar>); }

selectImages(images) {

this.setImages(images);

}

render() {

return (

<Content

<Form> <Item>

<Input placeholder="Report Title"

value={this.state.title}

onChangeText={value

=> this.setState({title:value})} />

</Item> <Textarea rowSpan={5}

bordered placeholder="Report Description"

value={this.state.description}

onChangeText={value

=> this.setState({description:value})} />

</Form>

<FlatList style={{margin: 5}}

data={this.getImages()} numColumns={2}

keyExtractor={(item, index) => `key\_${index}` }

renderItem={({item}) =>

<Card style={{ flex:1}}> <CardItem>

<Body

style={{alignItems:

'center',justifyContent:'center'}}>

<Thumbnail square source={item} />

</Body> </CardItem> </Card>

} />

<Button success full rounded onPress={this.setModalVisible.bind(this,true)}>

<Text> Add Screenshot </Text> </Button>

<View style={{ paddingTop:10}}>

<Button padding success full rounded onPress={this.submitReport.bind(this)} disabled={this.state.loading}>

<Text> Submit </Text> </Button> </View>

<Modal animationType={'slide'}

transparent={false}

visible={this.state.modalVisible}

onRequestClose={() => {

this.setModalVisible(false); }}>

{this.renderNavBar()}

<CameraRollPicke maximum={10}

imagesPerRow={4}

callback={this.selectImages} selected={[ ]} />

</Modal> </Content>

);

}

}

const styles = StyleSheet.create({

});

export default Report;

**General Announcements**

import React, { Component } from 'react'

import { View, StyleSheet, FlatList } from 'react-native'

import { Container, Content ,List, ListItem, Thumbnail, Left, Body, Right, Text, Button, Separator } from 'native-base'

import axios from 'axios';

import {Actions} from 'react-native-router-flux';

export class Announcement extends Component { constructor(props) {

super(props);

this.state = { data\_n: [],data\_p: [] }

}

previewContent(item){

Actions.preview\_announcement({

content:item

});

}

componentDidMount(){

axios.get('api/announcement').then( result

=> { console.log(result.data.priority);

this.setState({

data\_n: result.data.normal,

data\_p: result.data.priority

});

}).catch( error => {

console.log(error.response)

})

}

render() {

return (

<Container style={styles.wrapper}>

<Content> <Separator bordered>

<Text>Priority</Text> </Separator>

<FlatList data={this.state.data\_p}

keyExtractor={(item, index) => `pending\_${item.id}`}

renderItem={({item,index}) =>

<List> <ListItem thumbnail> <Left>

<Thumbnail square source={require('../../assets/images/da.png')} />

</Left><Body><Text>{item.title}</Text>

<Text note numberOfLines={1}>{item.description}

</Text></Body><Right><Button transparent

onPress={this.previewContent.bind(this,item)}>

<Text>View</Text></Button></Right>

</ListItem></List> } /><Separator bordered>

<Text>Normal</Text> </Separator>

<FlatList data={this.state.data\_n}

keyExtractor={(item, index) => item.id}

renderItem={({item}) =>

<List> <ListItem thumbnail>

<Left>

<Thumbnail square source={require('../../assets/images/da.png')} />

</Left> <Body>

<Text>{item.title}</Text>

<Text note numberOfLines={1}>{item.description}</Text>

</Body> <Right><Button transparent

onPress={this.previewContent.bind(this,item)}>

<Text>View</Text> </Button> </Right>

< /ListItem> </List>} /> </Content>

</Container>

)

}

}

const styles = StyleSheet.create({

wrapper: { flex: 1 },

searchWrapper: {

padding: 5 }

})

export default Announcement

**Appendix E**

**User Guide**

**Software Installation**

* Install BISU Information Hub App on your phone by tapping BISU Information Hub App.apk

**Running and Operating the Program**

* Tap the BISU Information Hub App on your phone.
* Log-in form will appear. Create your account and login to proceed. You won’t be able to use the app without an account. Inputting for the username and password is needed in accessing the system.
* After the user log in to the application tab menu scene appears and tap the tabs you want to trigger.

**RESEARCHER’S BIODATA**



**PERSONAL INFORMATION**

Name : Kent M. Hipos

Nick Name : “Kentoy”

Date of Birth : March 16, 1999

Home Address : Oy, Loboc, Bohol

Email Address : kenthipos@gmail.com

Civil Status : Single

Citizenship : Filipino

Religion : Roman Catholic

Gender : Male

Name of Mother : Marcelina H. Soria

Name of Father : Venancio M. Soria

**EDUCATIONAL BACKGROUND**

Elementary Education : Tambis Elementary School

Secondary Education :Oy National High School

Tertiary Education : Bohol Island State University, Balilihan Campus

Course : Bachelor of Science in Information Technology

Motto : “Success is not a key to happiness, Happiness is a key to

succes”.

**RESEARCHER’S BIODATA**



**PERSONAL INFORMATION**

Name : Jonmar U. Dacuba

Nick Name : “Mar, Ondong”

Date of Birth : March 06, 1999

Home Address : Boctol, Balilihan, Bohol

Email Address : dacubajonmar@gmail.com

Civil Status : Single

Citizenship : Filipino

Religion : March of Faith

Gender : Male

Name of Mother : Rogaciana U. Dacuba

Name of Father : Nestor B. Dacuba (deceased)

**EDUCATIONAL BACKGROUND**

Elementary Education : San Roque Integrated School

Secondary Education : San Roque Integrated School

Tertiary Education :Bohol Island State University, Balilihan Campus

Course :Bachelor of Science in Information Technology

Motto : “Have faith in God no matter what happened, trust Him”

**RESEARCHER’S BIODATA**



**PERSONAL INFORMATION**

Name : Christian F. Bitas

Nick Name : “Xtian”

Date of Birth : October 17, 1998

Home Address : Catigbian Bohol

Email Address : xtian@gmail.com

Civil Status : Single

Citizenship : Filipino

Religion : Roman Catholic

Gender : Male

Name of Mother : Susana P. Bitas

Name of Father : Venancio L. Bitas

**EDUCATIONAL BACKGROUND**

Elementary Education : Ambuan Elementary School

Secondary Education : Immaculate Mary Academy

Tertiary Education : Bohol Island State University, Balilihan Campus

Course : Bachelor of Science in Information Technology

Motto : “Honesty Is the Best Policy”.