HOSTELCOT IOS APPLICATION REPORT

1. PURPOSE

The Hostelcot app is used to efficiently find the hostel around based on the location. This app facilitates the users by providing an easy to interact UI. The hostel page would give detailed description about the facilities that it provides along with images and videos. In addition to the hostel search, the app supports a significant search feature that gives details about the place where the hostel is located. Details include listing out essentials like restaurants, malls, gyms etc. The use of google maps to will help to locate these details along with giving us the navigation and distances. App will allow students to post their requirements and also allow owners to post the availability

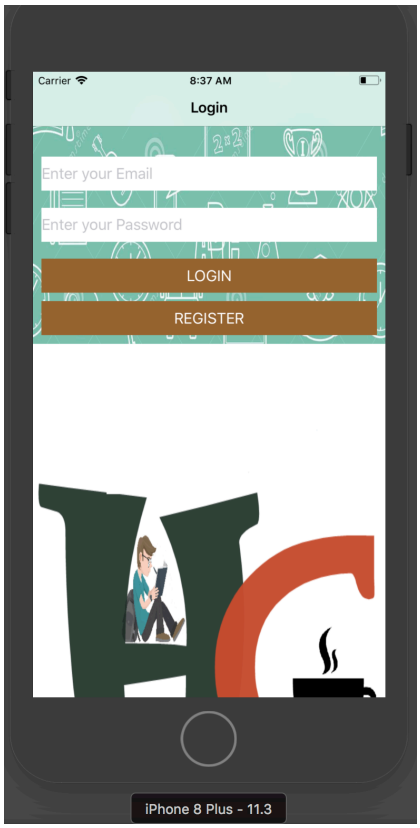
2. FEATURES

- Refined Searching of hostels on the basis of locations
- Efficient additional feature search for nearby landmarks like malls, hotels, gyms etc.
- Map display for all these places to propagate navigation
- Requirement Posting feature available for both students
- Availability posting feature for owners
- Detailed descriptive page for each hostel to highlight all its facilities along with visuals like images and videos

3. APP DETAILING

3.1 LOGIN

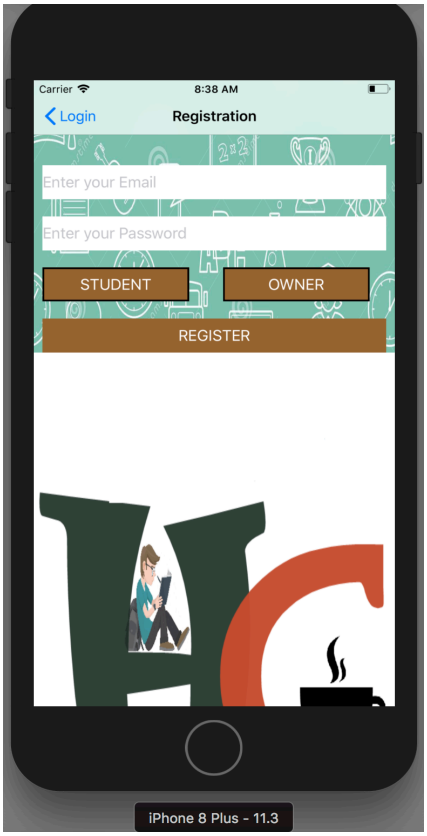
- User login on the basis of email and password



3.2 REGISTRATION

-User registration on the basis of email, password and user type

- User type can be student or owner



3.3 HOME

-Used tab bar controller to show multiple tabs

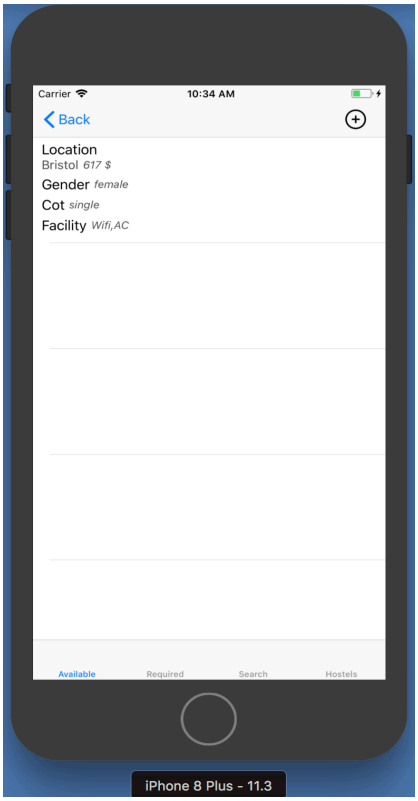
3.3.1 AVAILABILITY

-Shows the list of all posts tagged as available

-These are available to book for students

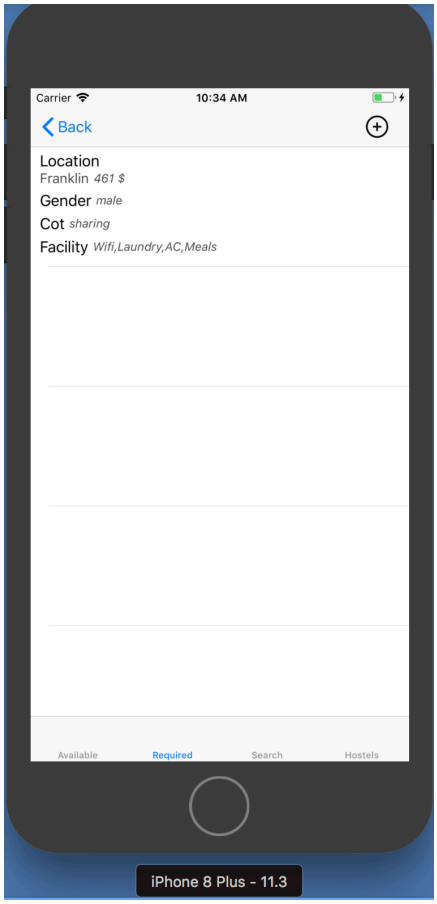
-These are posted by hostel owners

-Shows location, gender type, cot type, facilities and rent range for available hostel booking



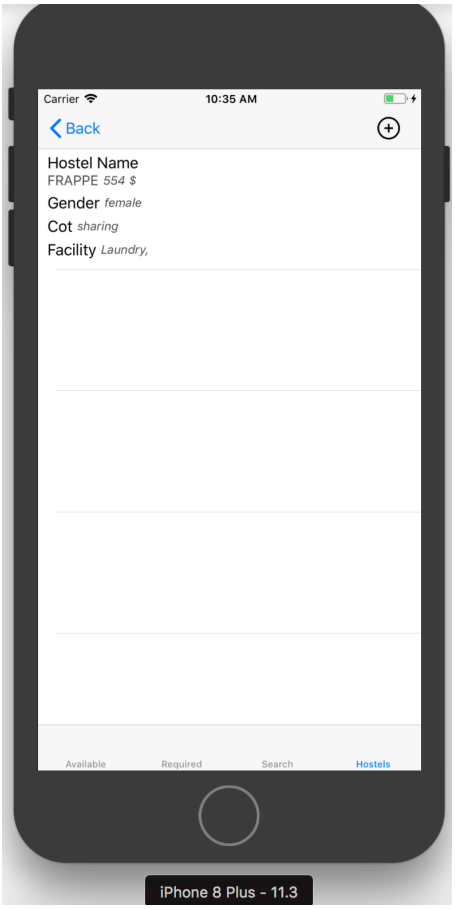
3.3.2 REQUIREMENTS

-Shows the list of all posts tagged as required
-These are required by the students
-These are posted by students
-Shows location, gender type, cot type, facilities and rent range for required hostel



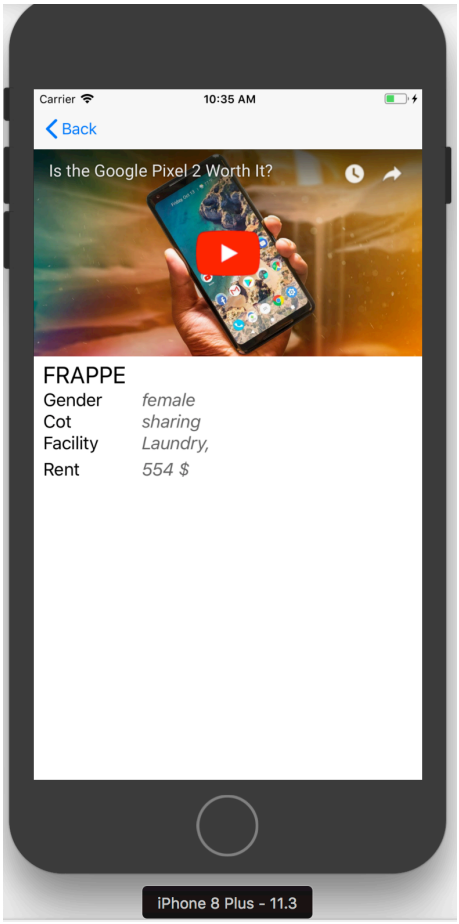
3.3.3 HOSTELS

-Shows the list of all hostels
-These are detailed views of hostels added to the app
-These are posted by hostel owners
-It shows the hostel page as detailed view for a hostel listing
-Shows names of hostels



3.3.4 HOSTEL PAGE

-A detailed page of any hostel
-Details given are hostel location, gender type, cot type, facilities and rent range
-Youtube video of hostel is also given
-Map location of hostel is also given
-These are created by hostel owners



3.3.5 SEARCH

-A filter page that allows user to set criteria for searching a hostel
-Filters given are hostel location, gender type, cot type, facilities and rent range
-On submission goes to search result page

3.3.6 SEARCH RESULTS

-Shows the list of hostels matching the given criteria

3.3.7 PROFILE

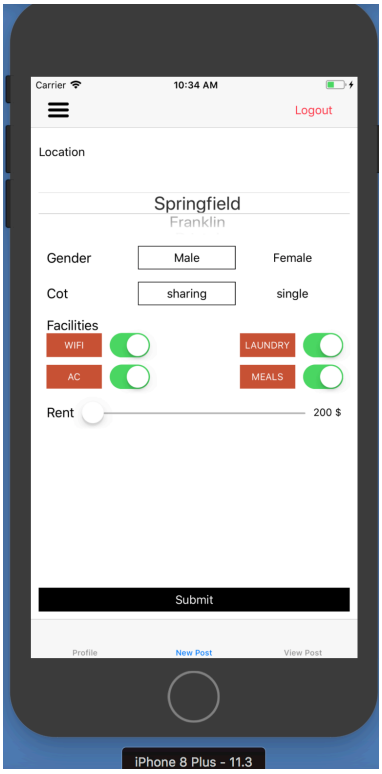
-Stores the phone, address details of a user

3.3.8 NEW POST

-Allows user to post a new listing

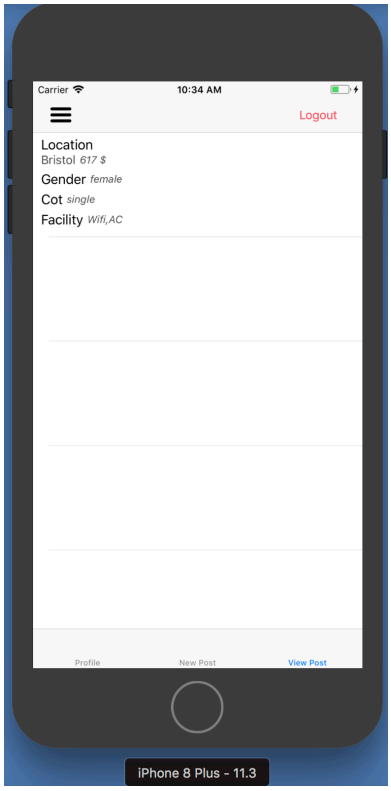
-If user type is student it is posted as requirement else if user type is owner its posted as availability

-Details given are hostel location, gender type, cot type, facilities and rent range



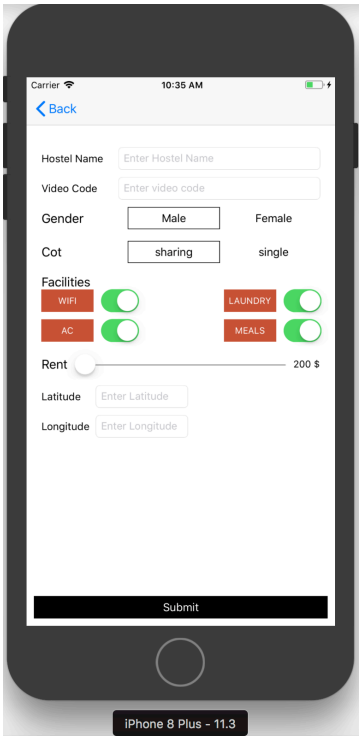
3.3.9 VIEW POSTS

-Shows the list of all posts created by a user



3.3.10 CREATE HOSTEL PAGE

-Allows owner to create a hostel page
-Details given are hostel location, gender type, cot type, facilities and rent range
-Youtube video of hostel is also given
-Map location of hostel is also given



4. PRIME FUNCTIONALITY

4.1 HOSTEL SEARCH

4.1.1 Search form

1) location = UIPicker
2) gender = UIButton
3) cot = UIButton
4) facility = UISwitches
5) rent = UISlider

4.1.2 Search Results

- Table view implementation of all results

4.2 AVAILABILITY POSTING

4.2.1 Post Form

1) location = UIPicker
2) gender = UIButton
3) cot = UIButton
4) facility = UISwitches
5) rent = UISlider

4.2.2 View Posts

- Table view implementation of all results

4.3 REQUIREMENT POSTING

4.3.1 Post Form

1) location = UIPicker
2) gender = UIButton
3) cot = UIButton
4) facility = UISwitches
5) rent = UISlider

4.3.2 View Posts

- Table view implementation of all results

4.4 HOSTEL PAGE

4.4.1 Create Page

1) location = UIPicker
2) gender = UIButton
3) cot = UIButton
4) facility = UISwitches
5) rent = UISlider
6) video = text box to input url
7) map = text box to input latitude and longitude

4.4.2 Page View

1) location = UIPicker
2) gender = UIPicker
3) cot = UIPicker
4) facility = UISwitches
5) rent = UISlider
6) video = Webkit view
7) map = Mapkit view

5. CODE SNIPPETS

5.1 Taking UIPicker value

```
func pickerView(_ pickerView: UIPickerView, didSelectRow row: In
    lbl_loc.text = pickerData[row]
}
```

5.2 Taking UISlider value

```
@IBAction func Act_rent(_ sender: UISlider) {
    rent = Int(sender.value)
    lbl_rent.text = "\(rent) $"
}
```

5.3 Taking UISwitch value

```
@IBAction func Act_wifi(_ sender: UISwitch) {
    if sender.isOn{
        iswifi = true
    }
    else{
        iswifi = false
    }
}
```

5.4 Taking UIButton value

```
@IBAction func Act_male(_ sender: Any) {
    btn_male.layer.borderWidth = 1.0
    btn_female.layer.borderWidth = 0.0
    str_gender = "male"
}
```

5.5 Checking User Type

```
if (UserDefaults.standard.object(forKey: "type") as! String) == "o
    self.ref = Database.database().reference().child("owner_post")
}
else{
    self.ref = Database.database().reference().child("student_post
}
```

5.6 Storing New Post

```
@IBAction func Act_submit(_ sender: Any) {
    let dict:NSMutableDictionary = NSMutableDictionary()
    dict.setValue(lbl_loc.text!, forKey: "loc")
    dict.setValue(str_gender, forKey: "gender")
    dict.setValue(str_cot, forKey: "cot")
    dict.setValue(isac, forKey: "ac")
    dict.setValue(islaundry, forKey: "laundry")
    dict.setValue(iswifi, forKey: "wifi")
    dict.setValue(ismeals, forKey: "meals")
    dict.setValue(rent, forKey: "rent")
    dict.setValue(UserDefaults.standard.object(forKey: "k_UID") as! String,

    arr_post.add(dict)

    print(arr_post)
    loader.start_loader(viewcontoller: self)

    self.ref.setValue(arr_post) { (error, reference) in

    }
}
```

5.7 Viewing the Stored Posts

```
override func viewWillAppear(_ animated: Bool) {

    arr_all_post.removeAllObjects()
    arr_usr_post.removeAllObjects()

    tblview.reloadData()

    loader.start_loader(viewcontoller: self)

    self.ref?.observe(.value, with: { (snapshot) in
        if let value = snapshot.value as? NSArray
        {
            self.arr_all_post = value.mutableCopy() as! NSMutableArray

            if self.arr_all_post.count > 0{
                for dic in self.arr_all_post{
                    let dict_host:NSDictionary = dic as! NSDictionary
                    if (dict_host.object(forKey: "uid") as! String) == UserDef
                        as! String{
                        self.arr_usr_post.add(dict_host)
                    }
                }

                self.tblview.reloadData()
                self.loader.stop_loader()
            }

        }
        else{
            self.loader.stop_loader()
        }
    })

}
```

5.8 Storing Profile Information

```
@IBAction func save(_ sender: AnyObject) {
    let dict:NSMutableDictionary = NSMutableDictionary()
    dict.setValue(tf_nm.text!, forKey: "name")
    dict.setValue(tf_phone.text!, forKey: "phone")
    dict.setValue(tv_add.text!, forKey: "add")

    loader.start_loader(viewcontoller: self)

    self.ref.setValue(dict) { (error, reference) in
        self.ref?.observe(.value, with: { (snapshot) in
            self.loader.stop_loader()
            self.navigationController?.popViewController(animated: true
        })
    }
}
```

5.9 Displaying Available Posts

```
        self.ref = Database.database().reference().child("owner_post")
        // Do any additional setup after loading the view.
    }

    override func viewWillAppear(_ animated: Bool) {
        arr_all_post.removeAllObjects()
        tableview.reloadData()

        loader.start_loader(viewcontoller: self)

        self.ref?.observe(.value, with: { (snapshot) in
            if let value = snapshot.value as? NSArray
            {
                self.arr_all_post = value.mutableCopy() as! NSMutableArray
                if self.arr_all_post.count > 0{
                    self.tableview.reloadData()
                    self.loader.stop_loader()
                }
            }
            else{
                self.loader.stop_loader()
            }
        })
    }
```

5.10 Displaying Required Posts

```
        self.ref = Database.database().reference().child("student_post")
    // Do any additional setup after loading the view.
}

override func viewWillAppear(_ animated: Bool) {
    arr_all_post.removeAllObjects()
    tblview.reloadData()

    loader.start_loader(viewcontoller: self)

    self.ref?.observe(.value, with: { (snapshot) in
        if let value = snapshot.value as? NSArray
        {
            self.arr_all_post = value.mutableCopy() as! NSMutableArray
            if self.arr_all_post.count > 0{
                self.tblview.reloadData()
                self.loader.stop_loader()
            }
        }
        else{
            self.loader.stop_loader()
        }
    })
}
```

5.11 Storing Hostel Page

```
@IBAction func Act_submit(_ sender: Any) {
    let dict:NSMutableDictionary = NSMutableDictionary()
    dict.setValue(tf_lat.text!, forKey: "lat")
    dict.setValue(tf_lon.text!, forKey: "long")
    dict.setValue(tf_hostel_nm.text!, forKey: "name")
    dict.setValue(tf_video_code.text!, forKey: "video")
    dict.setValue(str_gender, forKey: "gender")
    dict.setValue(str_cot, forKey: "cot")
    dict.setValue(isac, forKey: "ac")
    dict.setValue(islaundry, forKey: "laundry")
    dict.setValue(iswifi, forKey: "wifi")
    dict.setValue(ismeals, forKey: "meals")
    dict.setValue(rent, forKey: "rent")
    dict.setValue(UserDefaults.standard.object(forKey: "k_UID") as! String, forK

    arr_post.add(dict)

    print(arr_post)
    loader.start_loader(viewcontoller: self)

    self.ref.setValue(arr_post) { (error, reference) in

    }
}
```

5.12 Displaying Hostel List

```
        self.ref = Database.database().reference().child("Hostel_List")
        // Do any additional setup after loading the view.
    }

    override func viewWillAppear(_ animated: Bool) {
        arr_all_post.removeAllObjects()
        tableview.reloadData()

        loader.start_loader(viewcontoller: self)

        self.ref?.observe(.value, with: { (snapshot) in
            if let value = snapshot.value as? NSArray
            {
                self.arr_all_post = value.mutableCopy() as! NSMutableArray
                if self.arr_all_post.count > 0{
                    self.tableview.reloadData()
                    self.loader.stop_loader()
                }
            }
            else{
                self.loader.stop_loader()
            }
        })
    }
```

5.13 Displaying Hostel Page

```swift
override func viewDidLoad() {
    super.viewDidLoad()

   print(str_video)
    Videocode(videoCode:str_video)

    name.text = str_nm
    gender.text = str_gender
    cot.text = str_cot
    facility.text = str_facility
    rent.text = str_rent
    // Do any additional setup after loading the view.
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
    // Dispose of any resources that can be recreated.
}

func Videocode(videoCode:String)
{
    let url = URL(string: "https://www.youtube.com/embed/\(videoCode)")
    VView.load(URLRequest(url: url!))
}
```