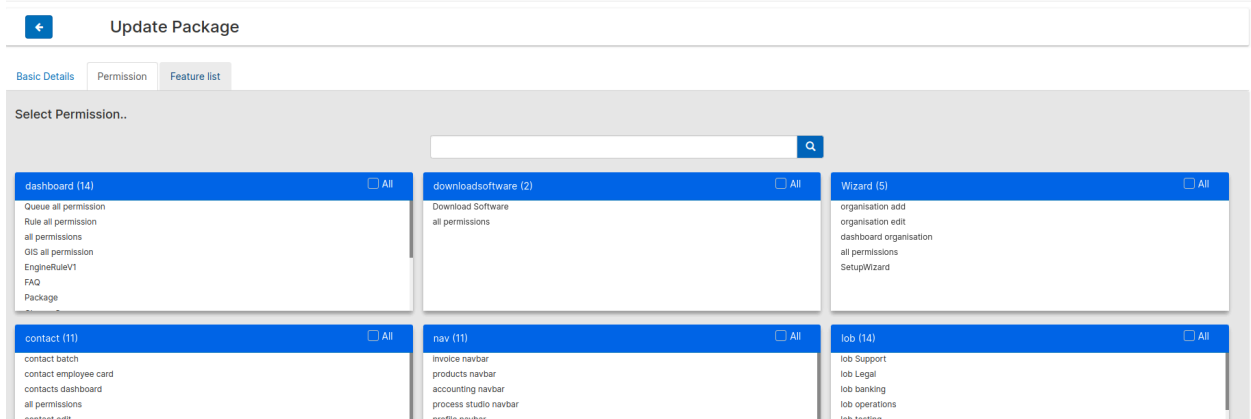


- User Price : If user need more userlimit then he can increase userlimit while purchasing package (so package price will increase with this user price / user)
 - While creating addOn package, have to click checkbox isAddOn and also have to link the addonpackage to main package.
- Also we can add permission for the package using the features tab.



- In FeatureList tab, we can add text message to show users while purchasing package.

Add Package :

- When user registers, he gets the free package (all the features & permissions in free package)
 - addpackage api is called from the registerUser api

```

let token = yield this.createTokenFor(String(userInfo._id),String(subscriberInfo._id),String(orgInfo._id));

let options = {
  method: 'POST',
  url: config.apiUrl.replace(/\/$/, '') + "/gibots-api/subs/addPackage",
  headers: {
    'authorization': "Bearer " + token,
    'content-type': "application/json",
    "selectedorgid": String(orgInfo._id)
  },
  body: [
    {
      "id" : packageData._id,
      "name": "Free Package",
      "amount": 0
    }
  ],
  json: true
};

process.env['NODE_TLS_REJECT_UNAUTHORIZED'] = '0';

let addPackage = yield this.remoteApiCall(options);

```

- And User can buy package manually through package type in their dashboard :



- After that these all packages are visible which can be purchased with checkout and buy features.
- Package related apis are in gibots-node-api. Productcontroller and subscribercontroller.
- In product document the features are stored with structure like this :

```

"features" : {
  "documentTypeList" : [
    {
      "name" : "Invoice_Document",
      "limit" : 50,
      "_id" : ObjectId("65f1c4fac400d9cfd823c5aa")
    }
  ],
  "processList" : [
    {
      "processName" : "Mini Agile",
      "limit" : 100,
      "chatLimit" : 20,
      "metaDataList" : [ ],
      "_id" : ObjectId("64e2f8003111793090b95c7d")
    }
  ],
  "reportList" : [ ],
  "expiry" : 45
},
"storage" : {
  "limit" : 20
},
"user" : {
  "limit" : 5
},

```

documentTypelist - Array of documentTypes
 processList - Array of processes
 reportList - Array of reports

- When we purchase any package, similar structure data is stored in subscriber document. In addition there startdate, expiry these things will be added.

```

    "pastPackages" : [ ],
    "features" : [
      {
        "startData" : null,
        "expiry" : null,
        "_id" : ObjectId("6618b756e71042b13f2c4816"),
        "documentTypeList" : [ ],
        "processList" : [ ],
        "reportList" : [ ],
        "packageId" : ObjectId("616d73d6ae22f84216e2501a")
      }
    ],
    "extraPurchase" : [ ],
    "storage" : {
      "limit" : 10,
      "used" : 0
    },
    "user" : {
      "limit" : 3,
      "used" : 1
    }
  },

```

- User can purchase multiple packages at a time, that's why features in subscriber document in in array format.
- If User purchases the same package again, then it goes the extrapurchase key of the subscriber and after expiry of the current package that package will automatically added in features array.
- In Subscriber document, there is one key called packages where we store all the packages which used for that subscriber. Using this packagesIds we fetch all permission from respective product document and give it to that user. So that he can use

```

    "amount" : 0,
    "cartList" : [ ],
    "packages" : [
      ObjectId("65e9bb686039c8ccb35ae82f"),
      ObjectId("62833ee8fd6b7508a62d5db7")
    ],
    "status" : "active",

```