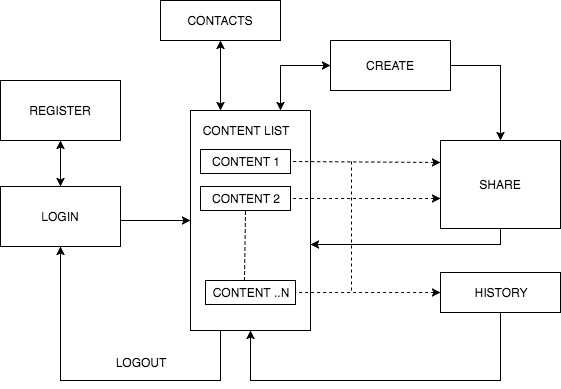
**bitYoga Documentation.**

bitYoga is an iOS app designed for sharing photos in encrypted way using blockchain technology.

**bitYoga Basic Navigation.**

****

# 

# 

# Screens.

## Login.

Login screen consists of two components *LoginScreen* and *Login*.

*LoginScreen* component is wrapper for the screen.

*Login* component implements the logic of the logic functionality.

### Functionality

* User can enter username and password to login.
* Register Link to navigate to register screen.
* Remember me check is present. If selected on next login user is navigated directly to contents page. If user uncheck remember me then user need to re-enter username and password.

### Components

#### Login

Method:

**goToLink()**: Hyperlink function to launch.

**clearErrorOnUpdate()**: Clear error info when user starts typing.

**onLogin()**: Login action. Calls Login API

**onLoginSuccess()**:

Success callback when user login. All info are stored in DB

**onLoginFailure()**:

Failure callback, when API call fails.

**checkData()**: Checks data is valid.

## Register.

Register screen consists of two components *RegisterScreen* and *Register*.

*RegisterScreen* component is wrapper for the screen.

*Register* component implements the logic of the register functionality.

### Functionality

* User can enter mobile, first name, last name, password to register.
* On successful register pop up shows registered successful message and navigate to login on closing it.
* Contain term and condition checkbox.\

### Components

#### Register

Methods:

**goToLink()**: hyperlink to Launch URL.

**checkData()**: check data is valid or not.

**onRegister()**: Register handle on Button tapped.

**clearErrorOnUpdate()**: clear Error prompts on user starts typing.

## 

## 

## Content List Screen.

List the posts viewable to the logged in user in chronological reverse order of the shared timestamp. This is an infinitely scrollable list of shared/created contents.History button leads to the corresponding content’s share history page where the user can see the the way the content is shared from it’s creator.

### Functionality.

* List out the content viewable to the logged in user.
* List keeps the chronological reverse order of share ie last got shared first.
* Infinite scrollable up to the first content that the user got shared.
* Contains links to the history, share, create, contacts.
* Contains logout, profile view functionality.

**Methods**:

**onSearchTextChange():** filter out the searchItems from the whole content list

**readuserInfo()**: read the user’s information stored in Local Storage

**processContents(content, resolve):** recieves the content from the block chain call and add to the existing list of contents in the current page’s state.

**Params:**

**content:** received content from the block chain call

**resolve:** resolve function to execute just after receiving the content obtained.

## Create Content Screen.

Provides feature to select an image to post, a message/title to type about the image selected. The degree of separation with the users that this content got shared is also specified in the create page. These informations are saved are shared automatically with the creator itself at the time of creation. The user is also asked to specify the contacts that the newly created content to be shared. The content created will be visible to the creator immediately since the content list is refreshed when the user navigated back to the content listing page.

### Functionality.

* User is able to select an image from his gallery.
* User is able to type a text about the image.
* User is able to specify the maximum degree of separation to the users who got shared the newly created content.
* The user is able to get shared the newly created content as soon as he navigate back to the content screen.

**Methods:**

**decrementShareCount():** decrementDegrees of Seperation value.

**incrementShareCount():** Increment Degrees of Seperation value.

**addImageClicked(options):**Triggers Gallery Open function.

**setImageDimension(image):** set’s the image’s dimension to fit the screen width without losing the aspect ratio.

Params:

**image**: selected image

**onDoneClick():**Triggers the block chain post call

**onShareClick():** Triggers the block chain post call followed by share post call.

**doPost(cb):** Common function that triggers the block chain post call followed by the triggered callBack function.

Params:

cb: callBack function

## Share History Screen.

Shows the tree of the shared history. From the root from the one who created the content(root) to all the way it shared to the person who never shared the content(leaf). If the user the person who created the content, he will have the option to forget the content so that no body can view the post anymore. Any user who got the content shared can view all the shared history as tree.

### Functionality.

* Shows the share history as tree.
* Creator will be able to do **forget content** .
* Creator will be able to revoke any user from the share history so that the revoked user no longer able to see the content.
* Any user who got shared the content can view all the user’s full name by pressing on the node.

**Methods:**

**getHistoryTree():** loads the transaction log in to the page. And also triggers the re data structure functionality.

**pushTreeNode(node)**: followup function to getHistoryTree to push the single transaction log entry in to the tree structure.

**nodeOnPress(node):** the functionassigned function onPress event on each tree node.

Params:

**node**: self pointer to the visible representing transaction log entry.

## Contacts Screen.

Contacts from mobile is read and check to bitYoga users list to find users.

Screen contain list contacts with action applied to them

Action are make friend, unfriend and send invite.

### Functionality.

* User can make friends by tapping ‘plus icon’ next to contact. This prompt for confirmation to execute action.
* User can unfriend an existing friend by tapping ‘ cross icon’ next to contact. This prompts confirmation to execute action.
* User can invite non existing contact by tapping ‘mail icon’ next to contact.
* Search bar is placed top of the list to filter the contact.
* To update the contact screen user has drag the contacts list downwards

### Components

#### ContactInviteScreen

**Methods**:

**Refresh()**: function to refresh contents of the screen.

**makeFriend()**: to make contact friend to user.

**unFriend()**: To remove friend from list.

**checkFriends()**: To structure data and finding friends from phone contact

**getContracts()**: To contact information about the contact from server.

**sortFriends()**: Sort friends based on status.

**\_refresh()**: wrapper for refresh function.

## Share Content Screen.

User can share contents direct from contents page. This is donwe using tapping ‘share icon’ in content.

### Functionality.

* Page contain back button to go back. ( top left corner )
* Information of content is displayed in this page like title, image, no of share left.
* Share Button is present to navigate to Share Friends Screen.

## Share Friends Screen.

In this page user can share content to select friends.

### Functionality.

* Share Button is share content to user is placed at top of the page.
* Select all checkbox is place at right to select all friends.
* Search bar is present to filter friend by the name.
* Select checkbox is place at right of the friend. Checking the box user is selected for sharing.

### Components

#### FriendShareScreen

**Methods**:

**shareContent()**: share content to single user.

**shareContents()**:

share content to multiple user. It recursivly calls shareContent func.

**getFilteredFriends()**: Filter friends by search string.

**onShareChange()**: on share Check action action to each friend in the list.

**onSelectAll()**: action to selectAll checkbox

## Drawer Screen.

This contain all the menu in the application.

### Functionality

* Contain Profile image at top. Current it is placed with placeholder image.
* Links
  + Profile: Link to profile page. Currently disabled.
  + Contacts: Link to contacts page.
  + Logout button: Link to logout of the application.
  + Help: Link to Help page. Currently disabled.