## User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| uid | bigint | No |  | auto\_increment |  |
| username | varchar(32) | No |  |  |  |
| email | varchar(64) | No |  |  |  |
| fname | varchar(16) | Yes | NULL |  |  |
| lname | varchar(16) | Yes | NULL |  |  |
| pwd | varchar(255) | No |  |  |  |
| activated | int(11) | No | 0 |  |  |
| avail\_storage | double | No | 250.0 |  |  |
| token | varchar(255) | No |  |  |  |

## Device

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| uid | bigint | No |  |  | User.uid |
| device\_id | int(11) | No |  |  |  |
| device\_name | varchar(255) | No |  |  |  |
| device\_system | int(11) | No |  |  |  |

## Sharing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| nid | bigint | No |  |  | Node.nid |
| shared\_uid | bigint | No |  |  | User.uid |
| date | datetime | No |  |  |  |
| expire | datetime | Yes | NULL |  |  |

## Node

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| nid | bigint | No |  | auto\_increment |  |
| uid | bigint | No |  |  | User.uid |
| device\_id | int(11) | No |  |  | Device.device\_id |
| nodename | varchar(255) | No |  |  |  |
| size | double | No |  |  |  |
| date | datetime | No |  |  |  |
| revision | int(128) | No |  |  |  |
| location | varchar(255) | No |  |  |  |
| permission | int(11) | No |  |  |  |
| pub\_link | varchar(255) | Yes | NULL |  |  |
| is\_thumbnail | smallint(6) | No | 0 |  |  |
| is\_folder | smallint(6) | No | 0 |  |  |
| is\_svnmeta | smallint(6) | No | 0 |  |  |
| parent\_id | bigint | No | 0 |  |  |
| has\_child | smallint(6) | No | 0 |  |  |

## System Design in terms of DropBox

The SyncGallery+ will interact with the backend online storage through the DropBox API. It will periodically send regular HTTP requests to retrieve images stored on the DropBox folder, and it will also upload the images from the Android phone to DropBox as indicated by the user. The gallery creates a DropBox album on the Android device and shows all uploaded images there. The DropBox album plays a role of public folder, sharing images among all users’ devices. Users can also share select images or albums with other DropBox friends. Once the user move the shared image out of the DropBox album, the image will get deleted from the online storage, but remain exist in the Android phone local storage.

## System Design in terms of Photo Editor

The SyncGallery+ is doing partnership with Aviary, so the backend engine of photo editor will be Aviary with little customization. In Gallery, after the user clicks “Edit” button, the SyncGallery+ will switch to the edit mode, which means, the Aviary will take over the rest of the work. Once the user click “Done”, the SyncGallery+ will switch back to the view mode, Aviary engine will be turned off.

## System Design in terms of Android Beam

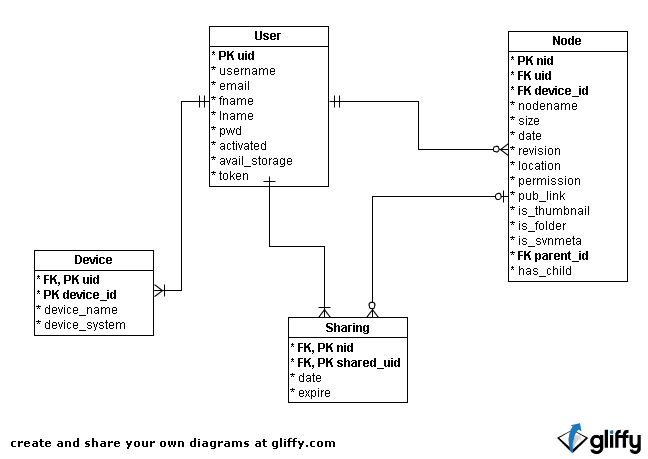
Android Beam is a new feature in the latest Android 4.0 Ice Cream Sandwich. It utilizes the NFC technology to transfer the data within a very short distance without any authentication processes. We believe this is a promising and convenient way to share the picture without the help of any social networking platform. Also, due to the reason that no authentication (pairing) process is involved, this is even faster than Bluetooth in practical. However, the transfer rate of NFC is pretty unacceptable relative to the size of image, and this is the reason why it is currently wildly used to transfer URL only. We are investigating a new way to share the image through Android Beam: To avoid transferring the large image file, we will first acquire a public link (expires in 5 mins, in most cases, it should be enough to finish the transaction) of the image with the help of DropBox from Android Beam issuer, and then send the public link from issuer to the receiver. Once the receiver gets the DropBox public link, downloads the image via 3G network or Wi-Fi and shows on the receiver’s phone screen. If 3G or Wi-Fi is not accessible, show a warning message and suggest the user to use Bluetooth to finish the transaction.

## System Design in terms of Social Networking

Like other photo gallery products, the SyncGallery+ allows the user to share the image via Social Networking platforms. We will support Google+ first in our milestone and Facebook later on. Without asking for the authentication information, we can easily retrieve the Google account information from the Android device, sharing the photos on Google+.

## System Design in terms of Error Handling

In SyncGallery+, 2 levels of errors will be mainly taken care of. Local validation will be implemented and run before the data send to the remote server in order to minimize the possibility of response failure. Remote validation is all done on the server side and we can’t interfere with its processing and return. Local validation will run again after the response data is received from the server. It ensures that there is no error occurred during the data manipulation on the server and the response data makes sense to the users’ requests. Users’ operation will also be monitored all the time by the Local validation system.



## Camera

The camera is one of the kernel features in SyncGallery+. With the camera embedded in our application, the user doesn’t have to jump out of the SyncGallery+ and take the photo by using other third-party applications. We are not going to spend too much time on polishing the camera functionality. As long as it can perform basic operations, we believe that it can fulfill the user’s requirement. Because SyncGallery+ is an application specially designed for photo editing, easy synchronization and easy sharing. Camera is just one of the methods to produce the image.

### Take a picture

The user can choose to switch to camera mode to take a picture within the application. In camera mode, we will only supply some basic functionality. To take a picture, the user simply taps the screen just as they do in stock camera application.

### Flash

Flash is also available in SyncGallery+. User can turn it on/off in the menu bar. By default, the flash is automatically managed by the system. In most cases, users don’t have to manually change its working status.

### Switch Camera

As most Android devices are now equipped with two cameras (one is in the front and the other is on the back), we decide to implement the functionality to allow the user to switch the working camera. By default, the camera on the back is considered as the major one the users commonly use.

## Social Network – Google+

Google+ is a social service launched on June 28, 2011. During the past half year, it has become the fastest growing social network platform in the world. On January 19, 2012, it was reported that Google+ had surpassed a user base of 90 million. Based on such a large number of user base and fast growing pace, we believe Google+ is a superb candidate platform that SyncGallery+ is supposed to support in first few milestones. Moreover, Android and Google+ are both products of Google. It could be much easier to integrate them together in practical development. We encourage SyncGallery+ users to be more active on Google+ as it introduces some creative ways to communicate, such as Circles. We are considering supporting more social network platforms in the future milestones, including Facebook and Flickr.

### 2.2 Share (Quick upload)

During the Android initialization steps, the user’s Google account information was already stored on the phone. Thus when the users choose to share the image on Google+, we can instantly retrieve the information from the phone device and finish the authentication process. The uploaded image will be shown in the “Quick upload” folder on Google+. 3G network or Wi-Fi is required to accomplish this process.

## Android Beam

Android Beam is identified as one of the most important new features in Android 4.0 Ice Cream Sandwich. It allows the rapid short-range exchange of bookmarks, contact info etc. via a near-field communication (NFC) protocol. The NFC communication has been more and more popular these years due to the fact that it requires an extremely simple setup only, replacing the pairing step of establishing Bluetooth connections or the configuration of Wi-Fi network. However, NFC only offers a low-speed connection relative to Bluetooth or Wi-Fi, which means it is hard to be used to transfer large size files, such as images. This is the main reason why there is no application of Android Beam on transferring the images in the app market. However, with the backend support from DropBox, we come up with an innovative solution to avoid the low transfer rate issue and make it possible to transfer the images in SyncGallery+. More technical details about this implementation can be found in System design – Android Beam section.

### Transfer Image via NFC

From the user perspective, the process works as if the real image file is actually sent via the NFC protocol. When the issuer Android device and the receiver Android device are pressed together, if NFC is available on both two devices, a message “Touch to Beam” appears on the screen. With on tap, the image is sent immediately to the receiving phone without any other operations. The receiving phone will show the image instantly on the screen in view mode.