LIVING LINEAGE PROJECT

**Lars comments/questions added**

**BACKGROUND**

Living Lineage is developing a social media website for families that will be private for approved members only. The content will be data from members that allows them to upload, add comments and tags, search for files, then retrieve and display them for the user. There will be many features added in the future but at this time we are only developing a test case for the most fundamental features for concept validation.

**BASIC ASSUMPTIONS**

* Data will consist of any type of digital file including photos, scanned documents, text files, audio and video files.
* Data needs to be stored separately from the website application with retrieval based on user requests, we can use AWS for data storage.
* Critical to the user experience will be speed, simplicity and an intuitive navigation throughout the entire platform. Ultimately, the data store will consist of millions of objects that need to be efficiently managed, queried and presented to users

**TECHNICAL MODULES**

1. **User**
	1. **User login system**
		1. Basic user login for now, needs password, username = email address
		2. Validate email address for first login
		3. Forgot password, reset password through email
		4. Multiple login attempts will lock the account and send an email to reactivate account
		5. Captcha can be added to stop spamming
		6. Admin can choose photos from backend to show on the public facing login screen.
	2. **Registration**
		1. Users should only be able to register through an invitation, no public registration on the home page.
			1. The admin or any user can send an invitation to an email address.
			2. The receiver clicks on a unique link in the email and lands on a “complete your registration” form
			3. The form asks user to input their credentials and set a password.
			4. Only admin can invite someone.
	3. User Profile
		1. User can manage their details like name, date of birth, place, address, email address, and password.
		2. Selection option of when to be notified when others add or comment to data – Sure. This can be immediate email or daily digest
	4. **Dashboard/Gallery view**
		1. After login user lands on the dashboard where all allowed images are shown in chronological/latest first.
		2. The gallery can be displayed in grid view like it is showing on Smugmug account.
		3. Data will be displayed by user selection priority
			1. Chronologically by date assigned to data
			2. By specific metadata hierarchy
			3. By location
		4. Files with audio associated will show a "speaker" icon alerting the user that audio content is available for that piece of data. Select "speaker" to play while viewing image
		5. Allow users to download selected files
		6. Clicking on a media item can be opened on a new page or in a popup window.
		7. In gallery view, the user can select multiple images by clicking on checkmark sign on top right and assign multiple tags at once.
		8. In top navigation/breadcrumb there will be quick gallery selection drop down
		9. All internal pages have a contact us button which will submit an email message to admin.
	5. **Search**
		1. User can query on any multiple metadata as well as use "and" and "or" functions
		2. Search can be auto-complete for the tags.
		3. Date range can be added in search
		4. "Add new search row" allows you to add another parameter. User can add as many parameters as required.
		5. Save-search feature.
		6. Select multiple images to download or download full album
		7. Add City, State, Country options to tags
	6. **Adding information**
		1. Users can comment and associate tags to others’ items.
		2. Text box available with each piece of data
		3. Previous edits shown with data
		4. Be able to edit/modify/add metadata
		5. Standard spell check available
	7. **Access rights**
		1. For now all users will have full edit access to all data, access authority will be introduced later but needs to be planned for now.
		2. User is assigned metadata as "owner" for their own uploaded data
		3. On top-right menu, users can access their own uploaded items.
		4. Only the owner can edit and delete an item.
	8. **Notifications**
		1. Once a new item is added, commented, edited then all relevant users will get notifications.
		2. The notifications can be shown on the top-right (similar to facebook) and the drop down menu will show link to each individual item.
		3. *Optionally* users can receive emails on every notification.
	9. **Uploading media**
		1. **File Formats**
			1. Photographs
			2. Videos
			3. Audio clips
			4. Scanned documents
		2. **Upload file process**
			1. Upload from computer or cell phone
			2. Upload progress bar
			3. Add metadata during upload. User supplied or data associated with the digital image such as date, time, location etc.
			4. Some meta data can taken from image’s EXIF data like date/time taken, place, etc.
			5. Add multiple metadata to multiple images at the same time, have "all" function. If 15 images are uploaded from a birthday party, be able to bulk tag all of them the same with person, dates, event etc.
			6. When uploading the user can select from existing metadata options and offer "new" in each category so they can create a new tag
				1. People, names
				2. Event
				3. Date
				4. Location
			7. Each Meta data is in fact a tag
			8. The very basic understanding of this schema is that Years, People, Places etc. are the main categories. For example of Years category; actual values of Years 2018, 2017, 2016… etc. are the Tags under this category. This is similar for People, places and Things. So admin will be able to define categories as well as tags under them like I showed you in the wireframes.
			9. Capture the metadata associated with the uploading image and save to db
			10. User can add more rows of tags/information like search “Add row”.
2. **Admin Panel**
	1. Admin dashboard is similar to user’s dashboard but with additional options
	2. Admin can update/delete any item, tag, comment and meta info.
	3. Tags management
		1. Add/Edit/Delete/Update
		2. If a tag is deleted, it will be dissociated from all photos
	4. Users management
		1. Add/Edit/Delete/Update/Ban
	5. App configuration
		1. Basic app configuration

**SITE LOOK, FEEL AND FUNCTION**

We are currently using SmugMug for image storage. The method and speed of presentation is nice but it lacks features we are looking for, it does not have a robust search function or ability for users to edit/comment

**PLATFORM**

The application will be built for web and mobile as a responsive web application.

**TECHNOLOGY**

* Language: PHP
* Database: MySql
* Front end: HTML5, CSS3, jQuery, Ajax, Angular.

**HOSTING REQUIREMENTS -**

* Platform: LAMP installation on a CentOS/Ubuntu server
* Hardware: A high speed VPS with at least 4GB of RAM and with root access will be good however dedicated server will be required down the road. The hardware can be upgraded at any time. The resource usage and load reporting will define this. There are many parameters like RAM or CPU usage goes to more than 80%, the disk is going to be full, number of requests etc.
* CDN: Media can be stored on a content delivery network. The disk space should be enough to accommodate the expected media upload.

# Project Schedule

|  |  |  |
| --- | --- | --- |
|  | **Tasks** | **Complete by** |
| Milestone 1 | Database design, Framework, |  |
| Theme HTML and layout |
| Milestone 2 | User Login, Registration, profile |  |
| User invitation, complete registration |
| Milestone 3 | Upload photo, associate meta data, multiple uploads, read exif data.  |  |
| Milestone 4 | dashboard, Galleries, breadcrumb, Collection view, Single view (photo, video, audio), navigation |  |
| Milestone 5 | Photo comments, add meta data to single item, download single item, Email notifications |  |
| Milestone 6 | Multiple items meta data addition, multiple downloads |  |
| Basic search |
| Advanced search with multiple tags |
| Milestone 7 | My uploaded media |  |
| Push Notifications |
| Milestone 8 | Admin panel for users management, tags management, site configuration |  |
| Milestone 9 | Testing |  |
|  |  |  |
|  | **Final Deployment** |  |