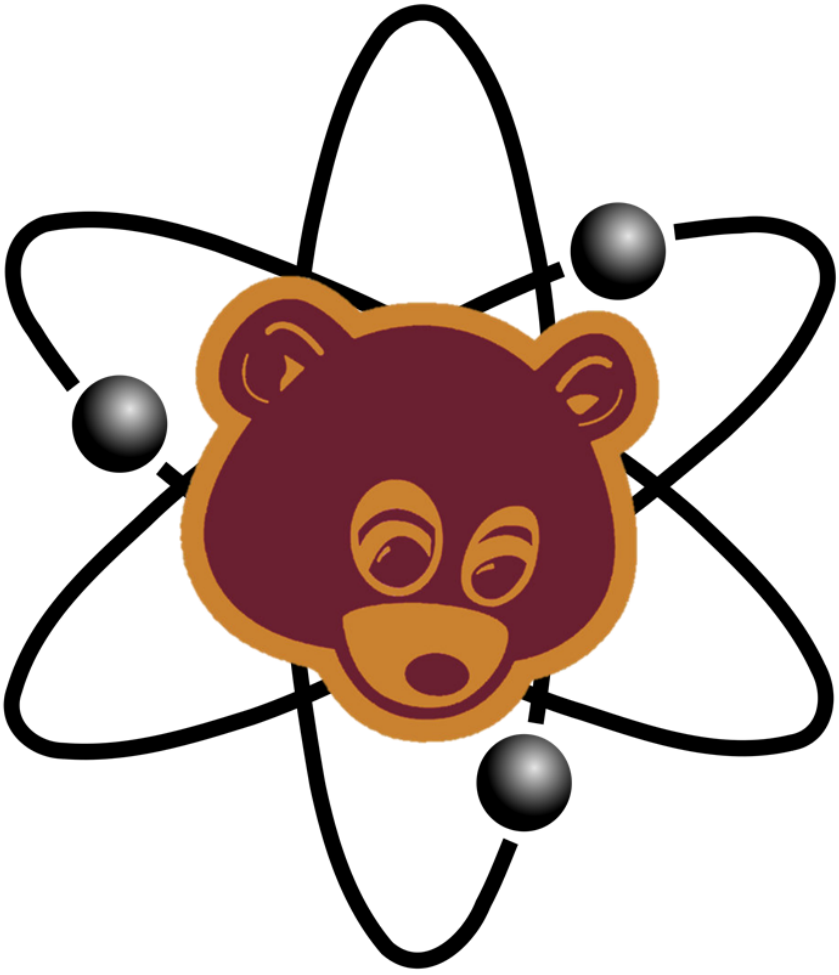
Molecular Playground Expansion



Team Dropout Requirements Specification

Table of Contents

1.0 Introduction

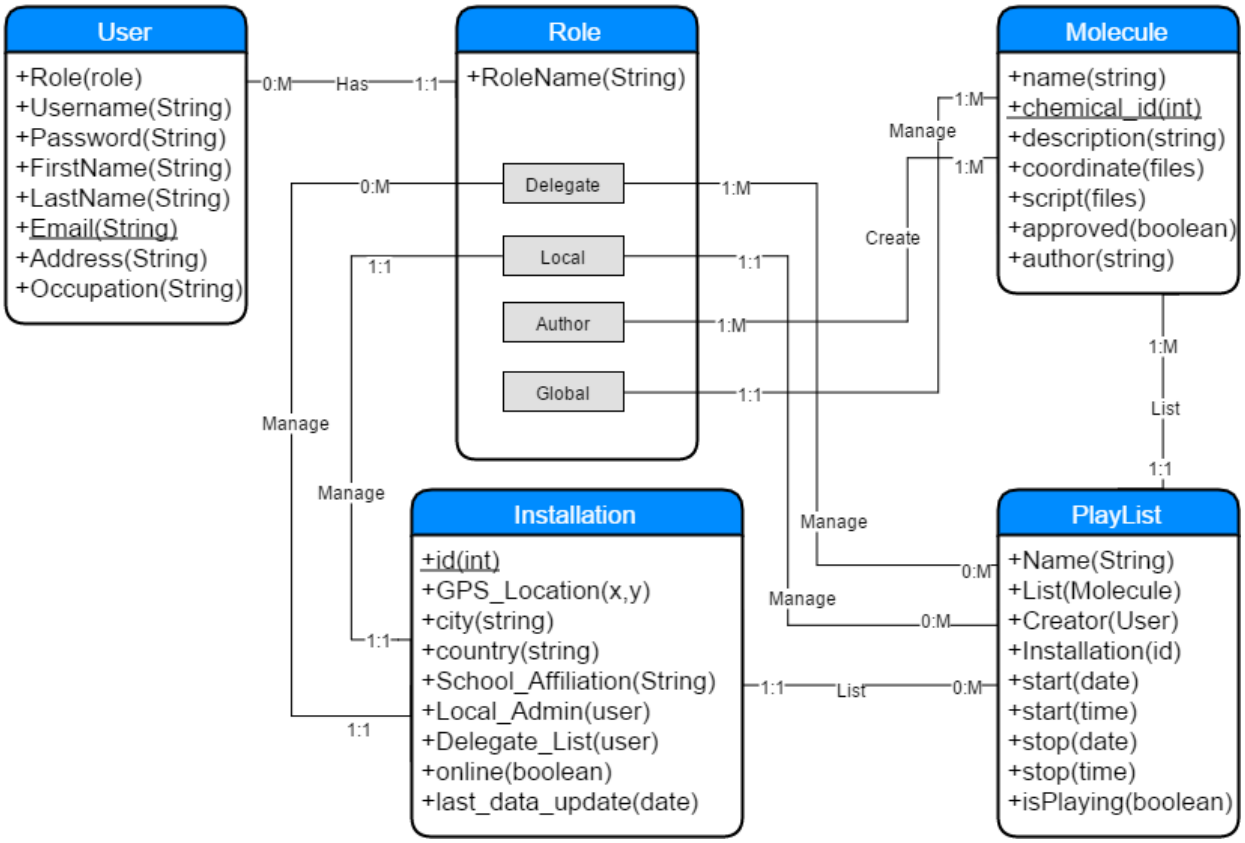
For a brief overview of the system's purpose continue reading Part 1 (Introduction). This section describes the existing Molecular Playground system and briefly explains what changes that are planned for the expansion. A reader of this document who wants to explore the Molecular Playground's new system's features more in depth should refer to Part 2 (Functional Requirements). In detail, that section covers all requirements and use cases of the Molecular Playground Expansion. Part 3 (Wireframes/GUI) is intended for the technical reader, for it shows the user interface and design plan. A reader interested in the goals for the new Molecular Playground Expansion should refer to Part 4 (Non-functional Requirements). That is the section that describes our goals regarding topics of performance, reliability, user privacies/restrictions, type of platform, portability, usability, scalability, and server details.

1.1 Background

Molecules are too small to see with the naked eye and the general public consequently views them as inaccessible and uninteresting. However, scientists know that the molecular basis of many macromolecules contains awe-inspiring beauty and induces great curiosity. In attempt to convey these ideas to the general public, the Molecular Playground was created.

The Molecular Playground is a system that displays large-scale interactive projections of molecules in public places. The equipment required for the system to function properly consists of a Kinect 360 Sensor, a computer with Mac OS X, the appropriate cables, and a screen or surface for the projection. The Molecular Playground is a working system with nine installments already in existence. However, the goal of the Molecular Playground Expansion is to improve and expand the already existing system to better suit the needs and wants of administrators, users and the client.

1.2 System ER Diagram



1.3 Function

The purpose of this document is to provide a concrete description of the Molecular Playground Expansion software requirements and features. Specifically, the changes being made to the existing system will be explained to ensure all desired requirements are fulfilled.

The long term goal of the Molecular Playground is to increase the amount of total installments around the world. To gain such a following, the software must become easier to maintain and access. Currently, each installation's data is stored on a local computer and any updates must be manually made. Our aim is to change the existing system so all playground files are stored on a central repository of data that can be accessed through the web. As the system updates, so should the local installments of the Molecular Playground.

Each local installment has a current playlist playing at all times when it's not being interacted with. Currently, a local administrator must create and manage a playlist through the local computer. The objective here is to change this and create a database of playlists on the central computer that a local administrator can select through the web. Once a playlist is selected, a local administrator should also be allowed to schedule which playlists are displayed on the Molecular Playground throughout the day.

Any new molecules can be submitted and viewed only by the global administrator. Once new content is reviewed and cleared, it can be uploaded to the site. With our system in place, this will make the new content available for selection by all local administrators.

1.4 Audience

The intended audience of this document includes:

1. Developer – to make sure all project developers agree on the basic features/requirements of the Molecular Playground Expansion and understand what changes are being made to the system
2. Investors - to make sure all interested investors understand the features and functionality of the Molecular Playground Expansion

The intended audience of this system includes:

1. Global administrators – to make sure the global administrator and primary client understands and agrees on the changes being made to the system and all of its features/requirements
2. Local administrators - to make sure all existing or future local administrators understand and agree on the changes being made to the system and all of its features/requirements
3. Delegates - to make sure all existing or future delegates understand and agree on the changes being made to the system and all of its features/requirements
4. General User - to make sure all interested users of the Molecular Playground Expansion understand the basic features/requirements of the new installment of the system

1.5 Scope

- All requirements our team will be developing in the Molecular Playground Expansion will be clearly explained and expressed in this document
- This document will not include any details for actual implementation
- Functional Requirements section will describe all possible use cases for each type of Molecular Playground Expansion user
- Wireframes/GUI section will present and examine the design plan for all parts of the Molecular Playground Expansion
- Non-Functional Requirements section will explain each of our goals for the Molecular Playground Expansion regarding performance, reliability, security, platform, portability, usability, scalability, and server details
- When the standards of the client is met, he/she will sign this document as an agreement to the given requirements for the new system

2.0 Glossary

Author: A user who can upload a molecule to the server.
Client: The person(s) who contracts terms for the development of new software attributable to the molecular playground.
Delegate: A user with the same permissions as a local administrator, except for the ability to manage other local delegates.
General User: A user who operates a local installation of the Molecular Playground
Global administrator (admin): The root administrator user who controls the installation setup and file approval.
Installations: Local system of the molecular interactive playground.
jMol: An open source java application used for viewing 3D chemical structures
jMol Script: The computer instruction file used to display the jmol rendering at every frame
Kinect 360 Sensor: Device used to allow users to interact with a molecule being projected.
Local administrator (admin): A user who manages the local molecular playground system.
Molecular Playground: The projection installation that shows the video rendering display.
Molecule: It is a virtual 3D representation of a particular molecule. In order for a user to display a Molecule, files need to be present that meet the format that jMol specifies. Molecules approved by global administrators will be shared with all global administrators, local administrators, and delegates.
Molecule List: The list of all molecule files the system currently contains.
New content: Molecule that local administrators uploads to the server.
Pending Request: After authors submit new molecule files. The global administrator will have these request appear in his pending request list for approval. Once it's approved it will be added to the system's database to share with other installations.
Playlist: List of molecules that are displayed in the order specified by this list. When playlists are created they are empty and molecules can be added and removed by the user creating this playlist. Playlists will be scheduled to play at the time specified by the local administrator or delegate. These are the pieces of information that a playlist requires name, list of molecules in playlist, creator of playlist, installation created at, start time, start date, stop date, stop time, and isPlaying.
Status: Status of server or database can be either online or offline and any numerical details such as last date data updated, number of videos played, and the playlist currently on..
User: A person who interacts with our system and falls under one of the four categories global administrator, local administrator, author, or delegate. Each user must have a role, username, password, first name, last name, email, address, occupation.
Web Server: Database that all installations will retrieve data from.

3.0 Functional Requirements and Use Cases

This section is broken down into five parts based off of the four different types of users our system can have and one section for functions all users in our system can perform. Global administrators, local administrators, delegates, and authors are the four types of users our system contains.

3.1 Global Administrator

Below are four functions that only global administrators can perform in our system.

Use Case ID:	3.1.1
Use Case Title:	View Status on Homepage (Figure 1.2)
Description:	Global administrator wants to view current status and how many installations are running, and highlight ones that are offline
Actors:	Global administrator
Pre-conditions:	Global administrator exist in the system
Post-conditions:	Display useful information about current system status
Trigger:	Login from the homepage
Normal Flow:	1. Displays the global administrator home page a. Map location address of the installations and current status of each. b. Status from database of usage, files, users
Alternate Flow:	
Exceptions:	1. Give status message if certain sections are not able to refresh

Created By:	Yong Liang	**Last Updated By:**	Freddy Nguyen
Date Created:	10/05/2015	**Last Revision Date:**	10/29/2015

Use Case ID:	3.1.2
Use Case Title:	Manage Molecule List (Figure 1.3a)
Description:	Global administrator wants to edit the molecule list
Actors:	Global administrator
Pre-conditions:	Global administrator is logged into the system
Post-conditions:	No changes to the database, give status message if certain sections are not able to save
Trigger:	Clicks the "Edit molecule list" section
Normal Flow:	1. Displays current list of molecules with options to make edits 2. Display pending requests for new molecule approval a. Global administrator can approve or reject 3. Confirmation box appears, and save on change
Alternate Flow:	
Exceptions:	3. If error, nothing will be saved, and it will respond with the error information

Created By:	Yong Liang	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/05/2015	**Last Revision Date:**	11/1/15

Use Case ID:	3.1.3
Use Case Title:	Manage Pending Requests (Figure 1.3b)
Description:	Global administrator manages pending molecule publication request
Actors:	Global administrator
Pre-conditions:	Global administrator is logged into the system
Post-conditions:	Approved molecule will be visible by all local installations
Trigger:	Clicks on “Pending Requests”
Normal Flow:	1. Pending requests are approved 2. Global administrator can leave a comment for user who created the request 3. Global administrator can save or cancel changes made to the requests
Alternate Flow:	1a. Pending Requests are rejected
Exceptions:	

Created By:	Freddy Nguyen	**Last Updated By:**	Main Khan
Date Created:	10/28/2015	**Last Revision Date:**	11/2/2015

Use Case ID:	3.1.4
Use Case Title:	Add New Installations (Figure 1.4c)
Description:	Global administrator wants to add an installation site
Actors:	Global administrator
Pre-conditions:	Global administrator is logged into the system
Post-conditions:	● Highlight each changed item and give status update on the top. ● Save changes to the database ● Both global administrator and local administrator get an email confirmation with access info to the installation
Trigger:	Clicks the add installation section
Normal Flow:	1. select “add new installation” 2. displays the form for add installation 3. enters the form information for the new installation 4. choose/generate username & password to setting up a local administrator 5. clicks the create button
Alternate Flow:	
Exceptions:	5. When not able to create, page is reloaded with the error message

Created By:	Yong Liang	**Last Updated By:**	Freddy Nguyen
Date Created:	10/05/2015	**Last Revision Date:**	10/29/2015

Use Case ID:	3.1.5
Use Case Title:	Manage Installations (Figure 1.4d)
Description:	Global administrator wants to edit existing installations and access
Actors:	Global administrator
Pre-conditions:	Global administrator is logged into the system, and the installation exists
Post-conditions:	● Disable or delete an existing installation ● Save changes to the database ● Both global administrator and local administrator get an email confirmation with changes to the installation
Trigger:	Clicks the "manage installation" section
Normal Flow:	1. select "edit an installation" from the list of current installations 2. next to each installation user can select update/ disable / delete a. update b. disable will lock access to the installation account c. delete will remove the installation 3. click save confirmed changes
Alternate Flow:	
Exceptions:	3. When not able to save edits, page is reloaded with the error message

Created By:	Yong Liang	**Last Updated By:**	Freddy Nguyen
Date Created:	10/28/2015	**Last Revision Date:**	10/29/2015

3.2 Local Administrator and Delegates

Local administrators and delegates share almost all of the same functionality except for three functions that only local administrators can do. For this reason section 3.3 contains functions that both the local administrators and delegates can do and section 3.4 contains functions only local administrators can perform.

Use Case ID:	3.2.1
Use Case Title:	View Playlists
Description:	Local administrator or delegate wants to view available playlists. (Figure 2.1b)
Actors:	Local administrator, delegates
Pre-conditions:	Local administrator or delegate is logged into the system
Post-conditions:	Lists of all playlists will be displayed
Trigger:	Select the "playlists" section in main menu
Normal Flow:	1. Select "playlists" from menu 2. Render playlists page, with list of playlists that have been created by current user
Alternate Flow:	2a. If current user has not created any playlists, the list of available playlists will be empty.
Exceptions:	2. Unable to load page from server displays error message

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/29/2015	**Last Revision Date:**	11/1/15

Use Case ID:	3.2.2
Use Case Title:	View Specific Playlist
Description:	Local administrator or delegate wants to view a specific playlist. (Figure 2.1d)
Actors:	Local administrator, delegates
Pre-conditions:	Local administrator or delegate is logged into the system
Post-conditions:	Specific playlist page is rendered
Trigger:	Select the “playlists” section in main menu
Normal Flow:	1. Select “playlists” from main menu 2. Render page displaying list of all playlists 3. Select the playlist to be viewed 4. Render the page for that specific playlist
Alternate Flow:	3a. If current user has not created any playlists, the list of available playlists will be empty. Selecting a specific playlist will be impossible.
Exceptions:	4. Unable to load information from server, display error message

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/29/2015	**Last Revision Date:**	11/1/15

Use Case ID:	3.2.3
Use Case Title:	Create Playlist
Description:	Local administrator or delegate wants to create a playlist. (Figure 2.1c)
Actors:	Local administrator, delegates
Pre-conditions:	Local administrator or delegate is logged into the system
Post-conditions:	● Save changes to the database ● Created playlist is instantly available
Trigger:	Clicks on the manage playlist section from menu
Normal Flow:	1. Select playlists from main menu 2. Render list of current playlists (empty if none created yet) 3. Select “create playlist” 4. New form is rendered, showing list of molecules to select/search from and the renderings for each molecule. 5. User selects the molecules that will be in the playlist 6. Select “save playlist” when finished 7. Render the newly created playlist
Alternate Flow:	
Exceptions:	6. No changes to the database, display error message that changes could not be saved

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/05/2015	**Last Revision Date:**	11/1/15

Use Case ID:	3.2.4
Use Case Title:	Edit Playlist
Description:	Local administrator or delegate adds and/or removes molecules from an existing playlist. (Figure 2.1e)
Actors:	Local administrator, delegates
Pre-conditions:	Local administrator or delegate is logged into the system, and at least one playlist has already been created and saved to database
Post-conditions:	● save changes to the database ● changes to playlist are immediately visible
Trigger:	Selects “playlists” from main menu
Normal Flow:	1. Display list of current playlists 2. User selects the desired playlist 3. Render selected playlist 4. User selects “edit playlist” button 5. Render list of molecules to add to playlist, and render option to delete each molecule that is currently in the playlist. 6. User selects molecule(s) to add to playlist 7. User selects “save changes” 8. Render edited playlist
Alternate Flow:	6a. User selects the “remove” button next to molecule(s) to remove from playlist.
Exceptions:	7. No changes to the database, display error message that items could not be downloaded

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/05/2015	**Last Revision Date:**	11/1/15

Use Case ID:	3.2.5
Use Case Title:	Schedule Content
Description:	Local administrator or delegate wants to schedule content to play at their site
Actors:	Local administrator, delegates
Pre-conditions:	Local administrator or delegate is logged into the system
Post-conditions:	● Save changes to the database ● Local installation receives schedule information to change what is playing
Trigger:	Clicks on schedule content option from menu
Normal Flow:	1. Render page that shows which playlist is currently playing on administrator's local installation, as well as option to change what is playing 2. Select new playlist to play 3. Render field to enter what time the playlist should start (current time is default). (Figure 2.4b) 4. Select "save" to submit timing information 5. Render success message with time that was entered
Alternate Flow:	
Exceptions:	4. Unable to save schedule information and/or unable to connect to the local installation. Display appropriate error message.

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/06/2015	**Last Revision Date:**	11/1/15

Use Case ID:	3.2.6
Use Case Title:	Upload Content
Description:	Local administrator delegate, or author uploads a new molecule, to be added to the molecule list upon approval by a global administrator. (Figure 3.3a)
Actors:	Local administrator, delegate, author
Pre-conditions:	Local administrator, author, or delegate is logged into the system
Post-conditions:	• File is sent to global administrator to await approval
Trigger:	Clicks on “upload” button
Normal Flow:	1. Render page for content uploading, as shown in (Figure 3.3a) 2. Local administrator chooses files to be uploaded (must be coordinate and script files, as specified) 3. Local administrator clicks on button to upload a molecule as seen in (Figure 3.3b) 4. Upload button is clicked to upload the files and await to be approved by a global administrator
Alternate Flow:	
Exceptions:	4. File upload fails due to poor connection, improper file format, or the molecule is too large.

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/8/15	**Last Revision Date:**	11/1/15

3.3 Local Administrator

Below are three functions that only local administrators can perform.

Use Case ID:	3.3.1
Use Case Title:	View Local Delegates
Description:	Local administrator wants to view list of delegates. (Figure 2.3a)
Actors:	Local administrator
Pre-conditions:	Local administrator is logged into the system
Post-conditions:	Page showing list of delegates is displayed
Trigger:	Select the “manage delegates” section in main menu
Normal Flow:	1. Render page showing list of delegate accounts that belong to current user’s installation
Alternate Flow:	1a. If current user has not created any delegate accounts yet, list of delegates will be empty.
Exceptions:	

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/29/2015	**Last Revision Date:**	11/1/15

Use Case ID:	3.3.2
Use Case Title:	Add Local Delegate
Description:	Local administrator wants to add local delegate. (Figure 2.3d)
Actors:	Local administrator
Pre-conditions:	Local administrator is logged into the system
Post-conditions:	● Changes saved to database ● New delegate can log in under their account created by local administrator ● Delegate is displayed in local administrator's list of delegates
Trigger:	Select the "manage delegates" section in main menu
Normal Flow:	1. Render list of delegates (Figure 2.3a) 2. Select "add delegate" option 3. Render and fill out account creation form (email, password, etc.) 4. Select "save delegate" 5. Local administrator and delegate receive email confirmation
Alternate Flow:	
Exceptions:	4. Changes are unable to be saved to database.

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/05/2015	**Last Revision Date:**	11/5/15

Use Case ID:	3.3.3
Use Case Title:	Edit Local Delegate Account
Description:	Local administrator wants to edit a local delegate's account. (Figure 2.3b)
Actors:	Local administrator
Pre-conditions:	Local administrator is logged into the system
Post-conditions:	● Changes saved to database ● Changes to delegate's account are shown immediately
Trigger:	Select the "manage delegates" section in main menu
Normal Flow:	1. Render list of delegates (Figure 2.3a) 2. Select delegate account from menu 3. Render account information form 4. Fill out fields in form that will be changed 5. Select "save delegate" 6. Local administrator and delegate receive email confirmation of changes
Alternate Flow:	4a. If local administrator wishes to remove delegate account completely, select "remove delegate" button
Exceptions:	5. Changes are unable to be saved to database.

Created By:	Ethan Miller	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/05/2015	**Last Revision Date:**	11/1/15

3.4 Author:

Below is a function that only authors can perform in our system.

Use Case ID:	3.4.1
Use Case Title:	Create Author Account
Description:	A user will create a new author account
Actors:	Author
Pre-conditions:	Author does not already have an account
Post-conditions:	A new author has created an author account
Trigger:	New Author clicks on “Sign Up” on homepage
Normal Flow:	1. Author is directed to sign up page (Figure 3.2) 2. Author fills in his desired Username and Password and selects his country and occupation 3. Author clicks continue and new author account is created
Alternate Flow:	
Exceptions:	3. Author picks a username that is already registered in our system and is asked to pick another username

Created By:	Main Khan	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/29/2015	**Last Revision Date:**	11/2/15

3.5 All Users:

The next three functions are ones that any of the four types of users in our system can perform.

Use Case ID:	3.5.1
Use Case Title:	Logging In
Description:	Any user logs in to the website with a username and password. (Figure 3.1)
Actors:	Global administrator, local administrator, delegate, author
Pre-conditions:	User has an account
Post-conditions:	User is logged in to the system
Trigger:	Open the website's login page
Normal Flow:	1. User enters username and password 2. The Web Server checks the user's credentials with user database and grants access if credentials are valid
Alternate Flow:	
Exceptions:	2. User credentials are incorrect, return user to log in page

Created By:	Freddy Nguyen	**Last Updated By:**	Matthew Lydigsen
Date Created:	10/05/2015	**Last Revision Date:**	11/1/15

Use Case ID:	3.5.2
Use Case Title:	Update Profile Information
Description:	User updates profile information (Figure 3.4)
Actors:	Global administrator, local administrator, delegate, author
Pre-conditions:	User is logged into the system
Post-conditions:	Return back to the home page after saved successfully
Trigger:	User clicks the menu bar “update account information”
Normal Flow:	1. Display page with the current user’s info and options to edit for each field 2. Click “Save” after all the edits
Alternate Flow:	
Exceptions:	2. Shows where the errors are on the top of the page and ask to re-enter those fields

Created By:	Yong Liang	**Last Updated By:**	Freddy Nguyen
Date Created:	10/05/2015	**Last Revision Date:**	10/29/2015

Use Case ID:	3.5.3
Use Case Title:	Logging Out
Description:	Any user will log out of the system. (Figure 3.1)
Actors:	Global administrator, local administrator, delegate, author
Pre-conditions:	User is logged in to the system
Post-conditions:	User is logged out of the system and redirected to the homepage (Figure 1.0a)
Trigger:	User clicks “Sign Out” on the sidebar
Normal Flow:	1. User is logged out of the system 2. User is redirected to the homepage (Figure 1.0a)
Alternate Flow:	
Exceptions:	

Created By:	Freddy Nguyen	**Last Updated By:**	Main Khan
Date Created:	10/05/2015	**Last Revision Date:**	10/29/15

4.0 Non-Functional Requirements

The non-functional requirements dictate our goals for program in terms of usage and application. This section will cover the user interface in terms of readability, usability, performance and general implementation specifications.

4.1 Offline Capabilities

a. If network fails, client can continue the use of their desktop application without the ability to interface with the Molecular Playground.
b. Scheduled playlist feature will work without network connection and will use the most recently scheduled playlist.

4.2 Platform

c. **Languages**
    i. We are in the process of determining the backend languages and database technologies we will use to implement the backend.
    ii. The web frontend will be written in HTML, CSS, and Bootstrap.
    iii. The user application is already written in C, C++, and Java, and we will be modifying it to include Molecular Playground functionality.
d. **Portability**
    i. The JMOL Java desktop application is cross-platform, and the web interface for the Molecular Playground will be compatible with multiple major browsers such as Chrome, Safari, and Firefox.

4.3 Usability

e. **Intuitive Usage**
    i. A user with no prior knowledge of the functionality of the program can intuitively manipulate the molecule using hand gestures.
f. **Advanced Users**
    i. People with knowledge of the functionality of the program and with the appropriate administrative rights can queue playlists, upload new molecules, etc.
g. **Handles Errors**
    i. The application will never throw an exception that causes the system to fail. It will be highly fault-tolerant.
h. **Provides Feedback**
    i. The system will provide feedback when something is successfully completed (e.g. queuing a task, reading/writing from the server, error messages, etc).
    ii. We will maintain a log of events of failure in the database to aid in the future fault-tolerance of the system.

4.4 Server

i. **Stores Molecules**
   i. The server will store the molecules created by authors and all accompanying information they provide.

j. **Server Capability**
   i. Many molecules can be pushed from many locations to the server at any time and it will handle these requests in efficiently.

4.5 Performance

k. **Maintains current performance time**
   i. Our implementation will run no slower than the current implementation.

5.0 Future Planning

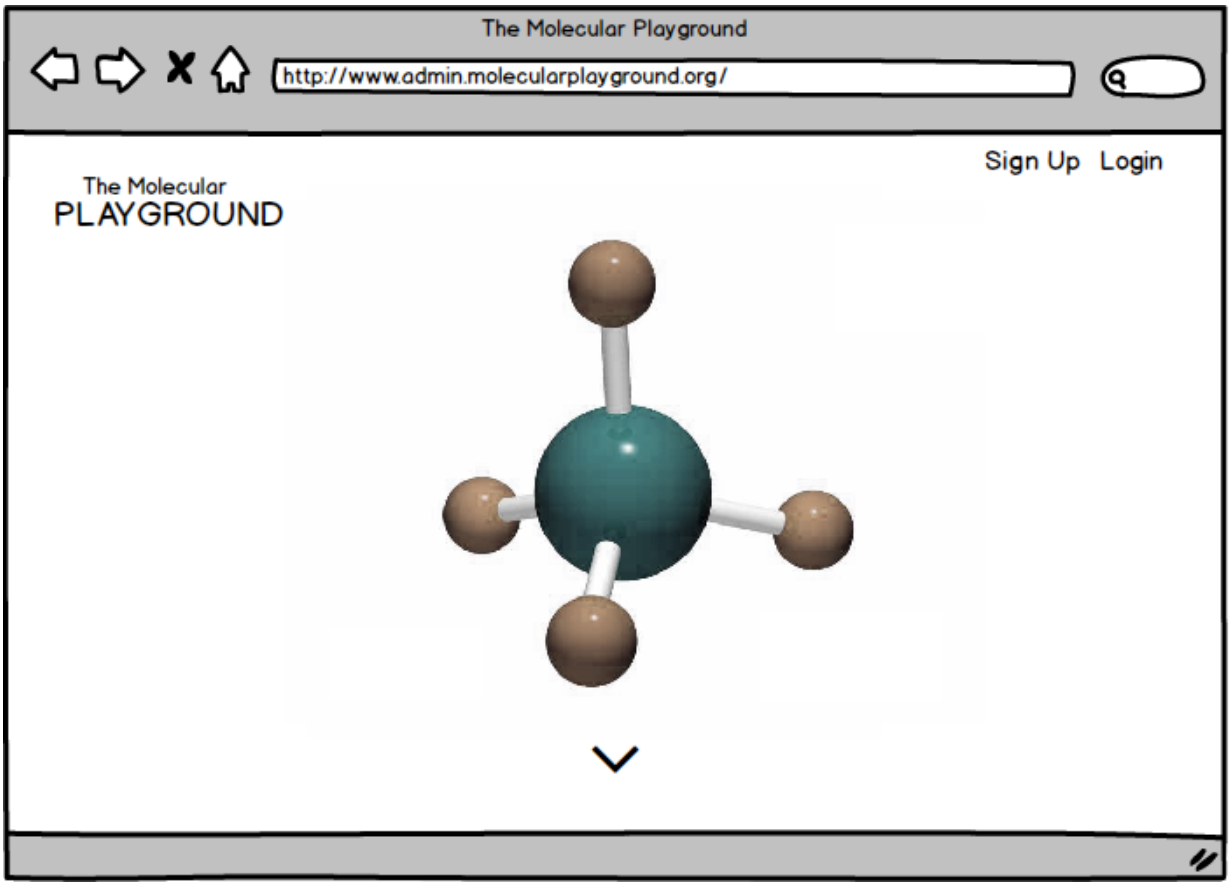
Because this class is a single semester commitment, there are inevitably some ideas that will not come to fruition as a result of our work this semester. In the future, some ideas that could be worth pursuing are:

1. Mobile Front-end Application

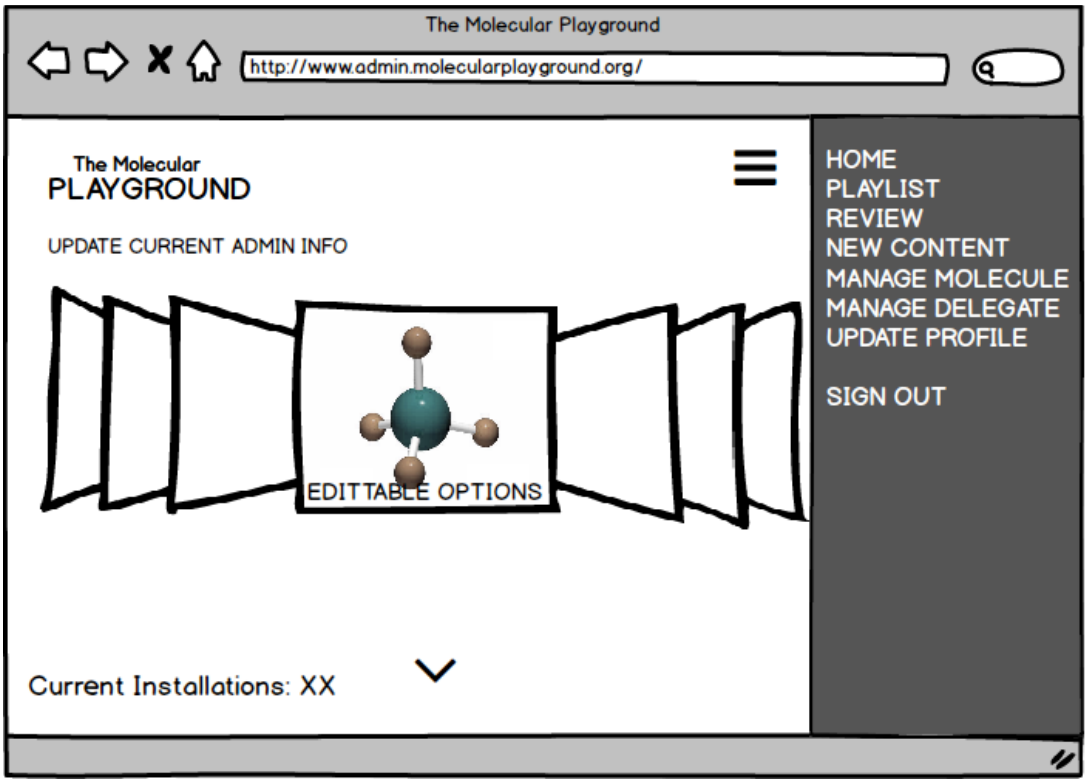
    a. Few (if any) of us have experience building mobile applications. We plan on using a web-service-oriented backend, so it should be as easy as building the UI to sit on top of our functionality for whoever ends up doing this
2. Increased server capabilities
    a. Having a single server to host all functionality could potentially be a problem. At this time, we do not have the resources to expand the system. Some possible solutions could be a public cloud, or simply more private computation power and storage.
3. Source Code Analytics
    a. Using a source code scanning tool such as Fortify or Pylint, scan the source code to deal with coding style inconsistencies and to find potential security vulnerabilities. While this would be a nice thing to do, we do not have the time to fix the existing system as well as our new one.
4. Molecule Statistics
    a. Offer statistics on the most downloaded, viewed, longest viewed molecules. Could potentially add the option to "Like" a molecule, perform statistics on that.
5. Molecule/jMol Upload preview
    a. Able to preview the files before confirming the upload. Possible implementing plugin or link to jMol.

6.0 Wireframes

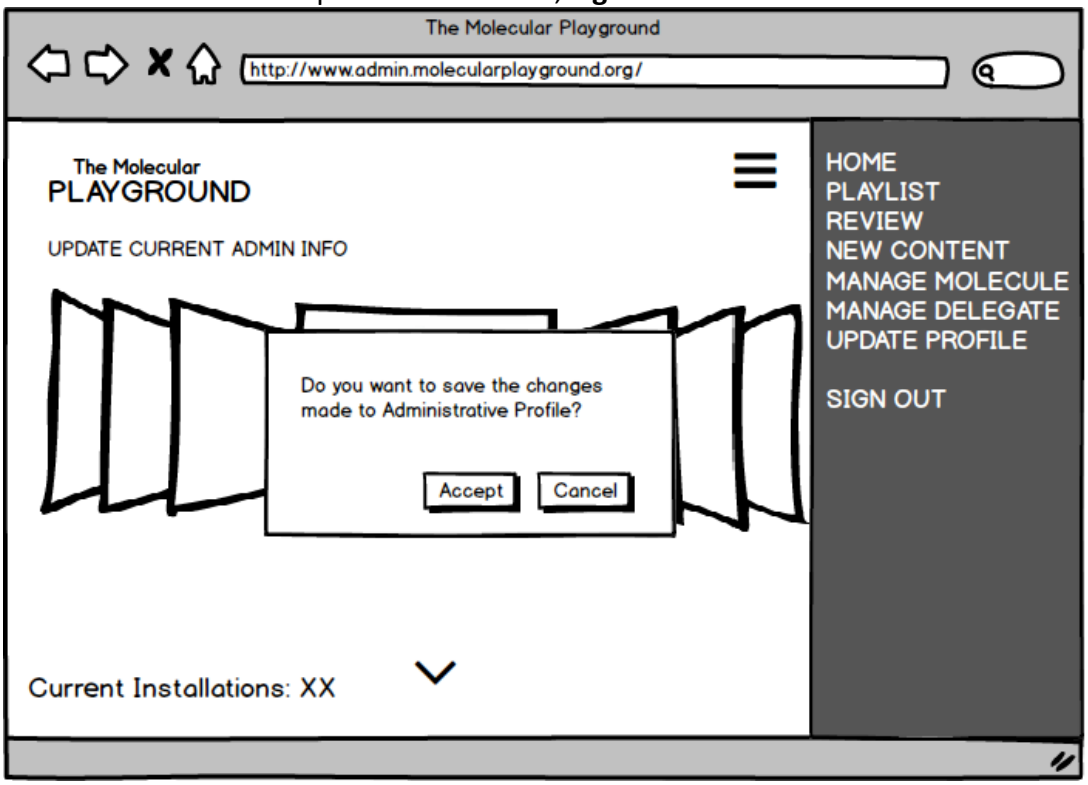
Listed below is a series of conceptual images depicting our vision for the extended application in development. These wireframes directly correlate to the use cases mentioned in section 3.0.



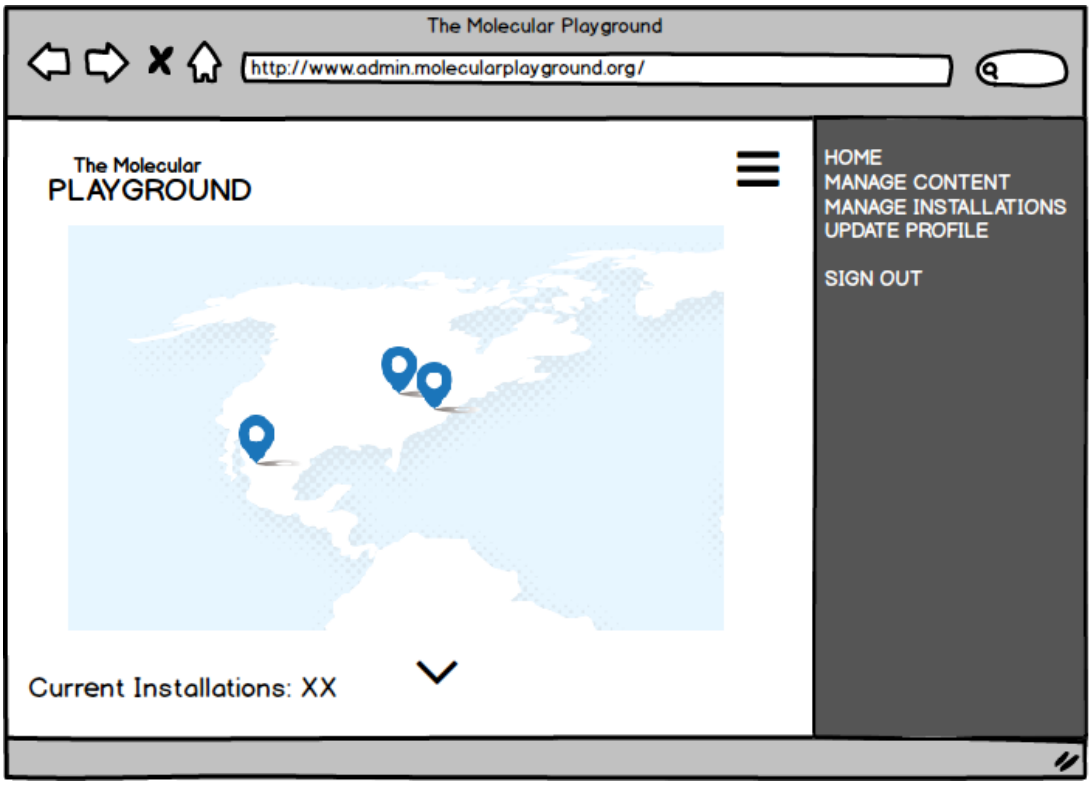
The front page before the user logs into the system
The Landing Page for the Web Application, **Figure 1.0a**



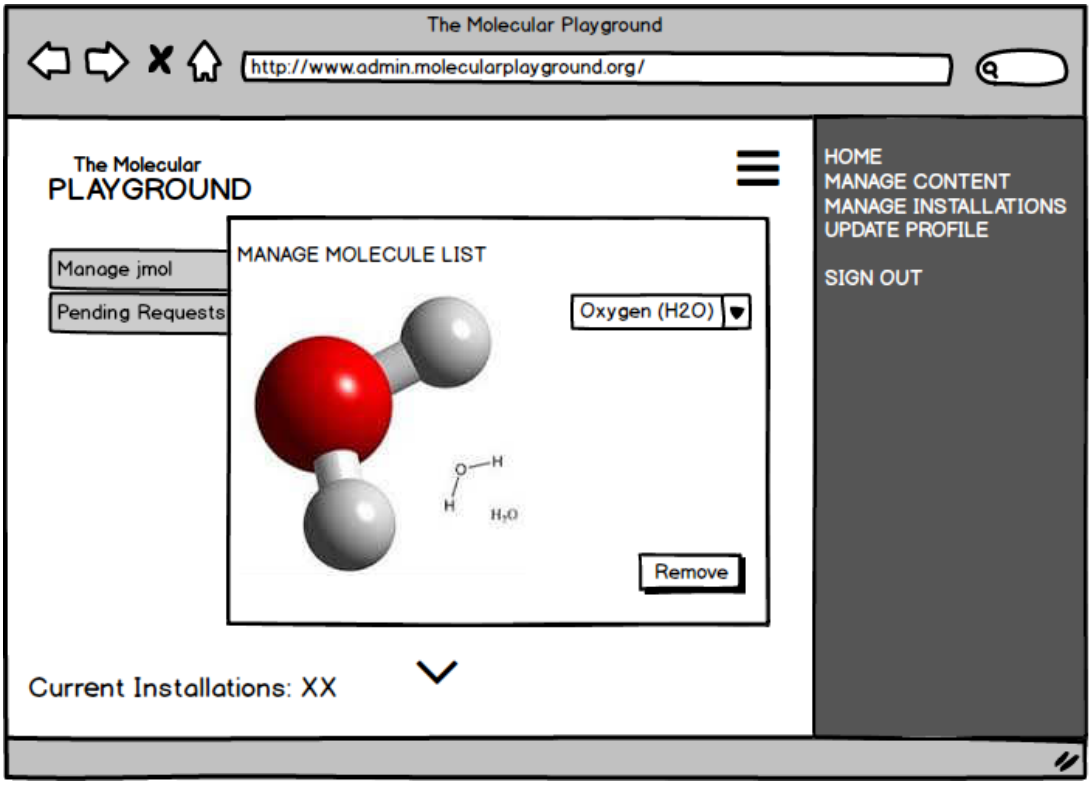
Global administrator - Update Information, **Figure 1.1a**



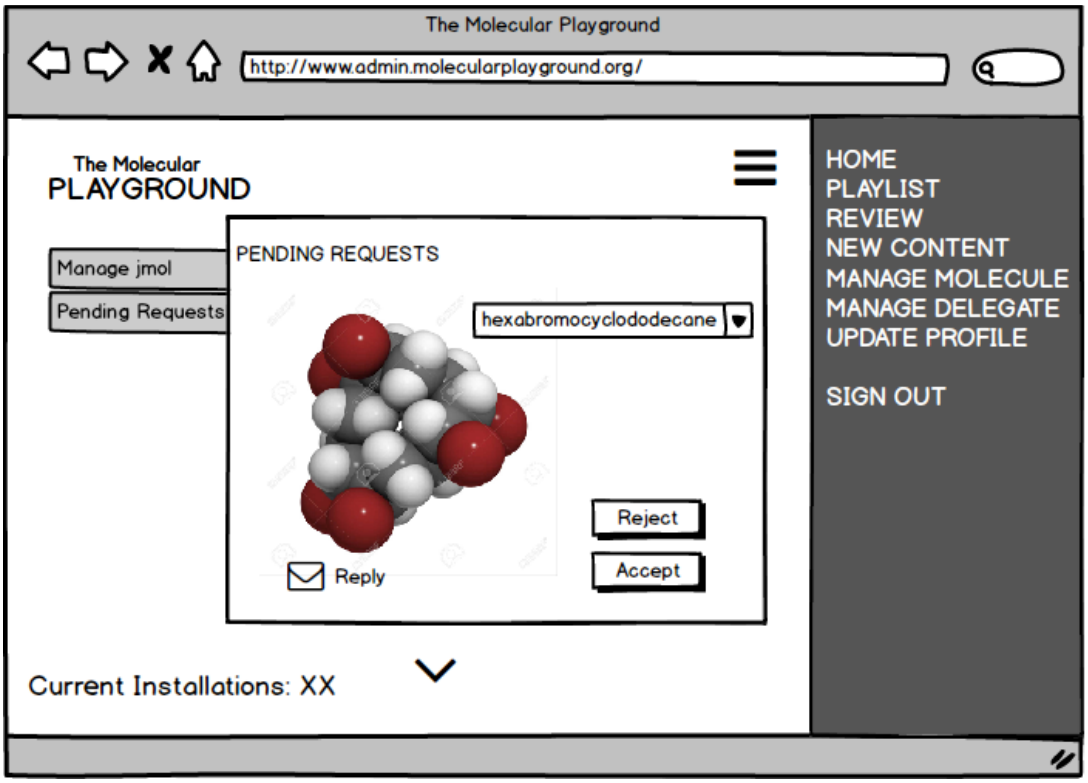
Global administrator - Update Information Prompt, **Figure 1.1b**



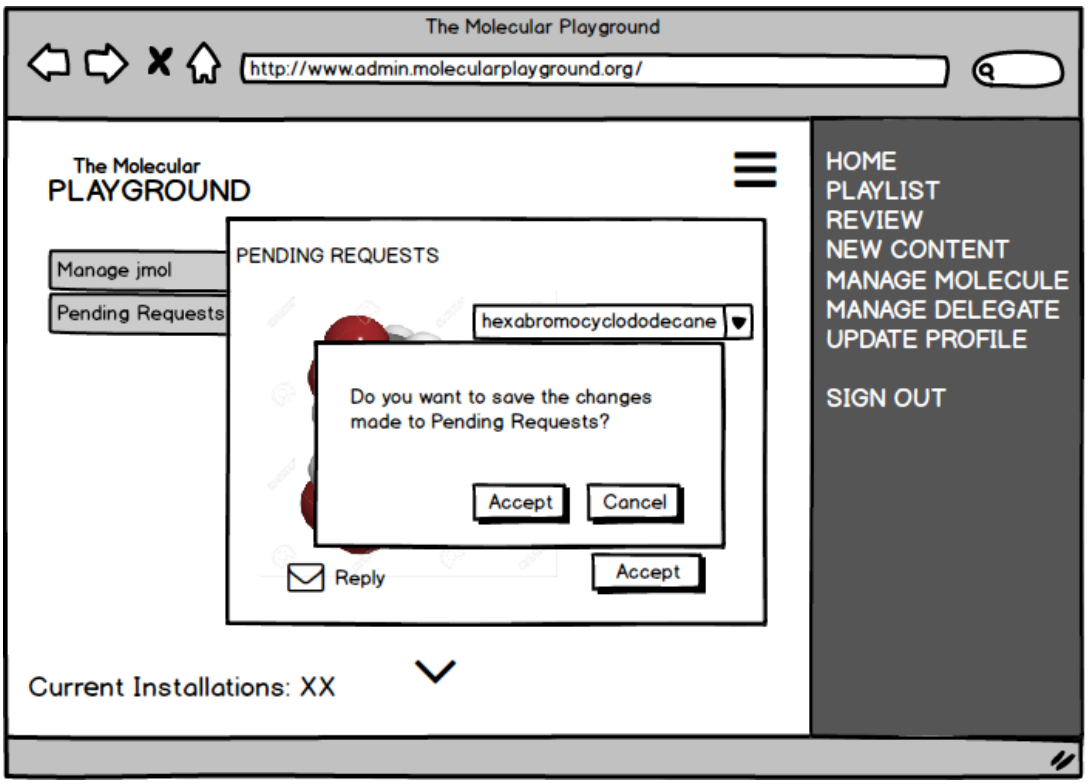
The homepage after the global administrator has logged into their account
Global administrator - Status View on Homepage, **Figure 1.2**



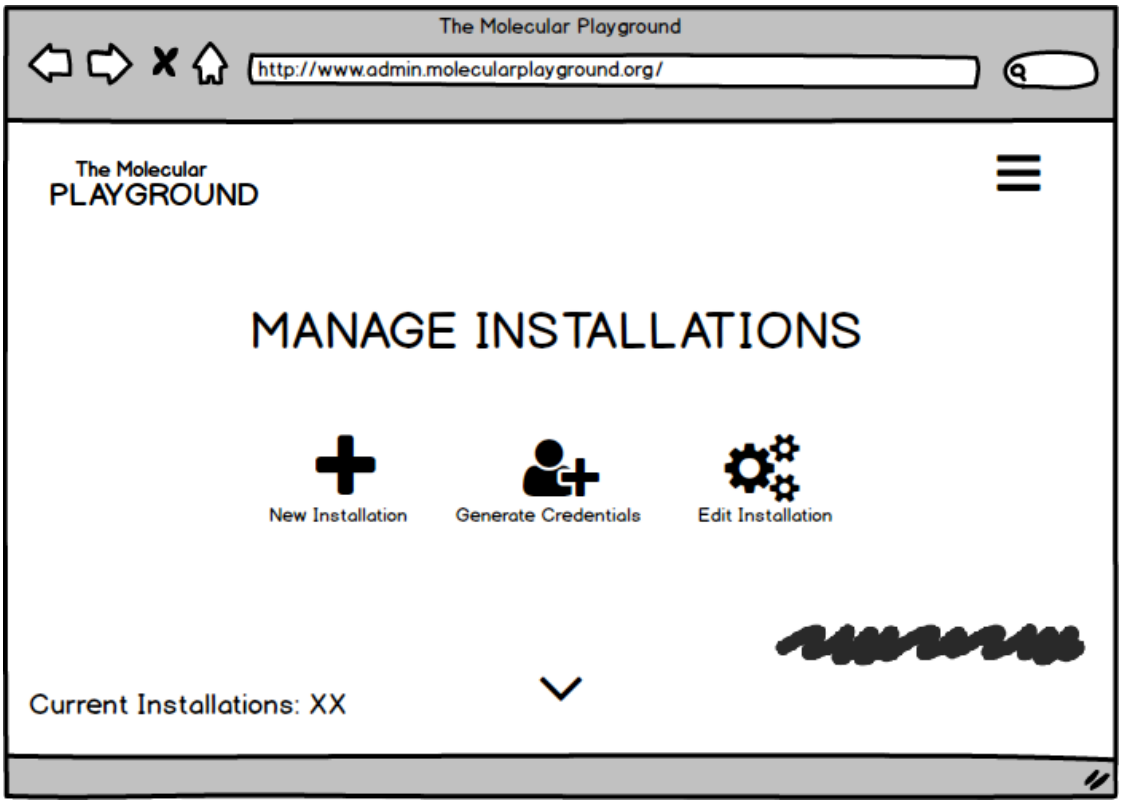
Global administrator - Manage Molecule List, **Figure 1.3a**



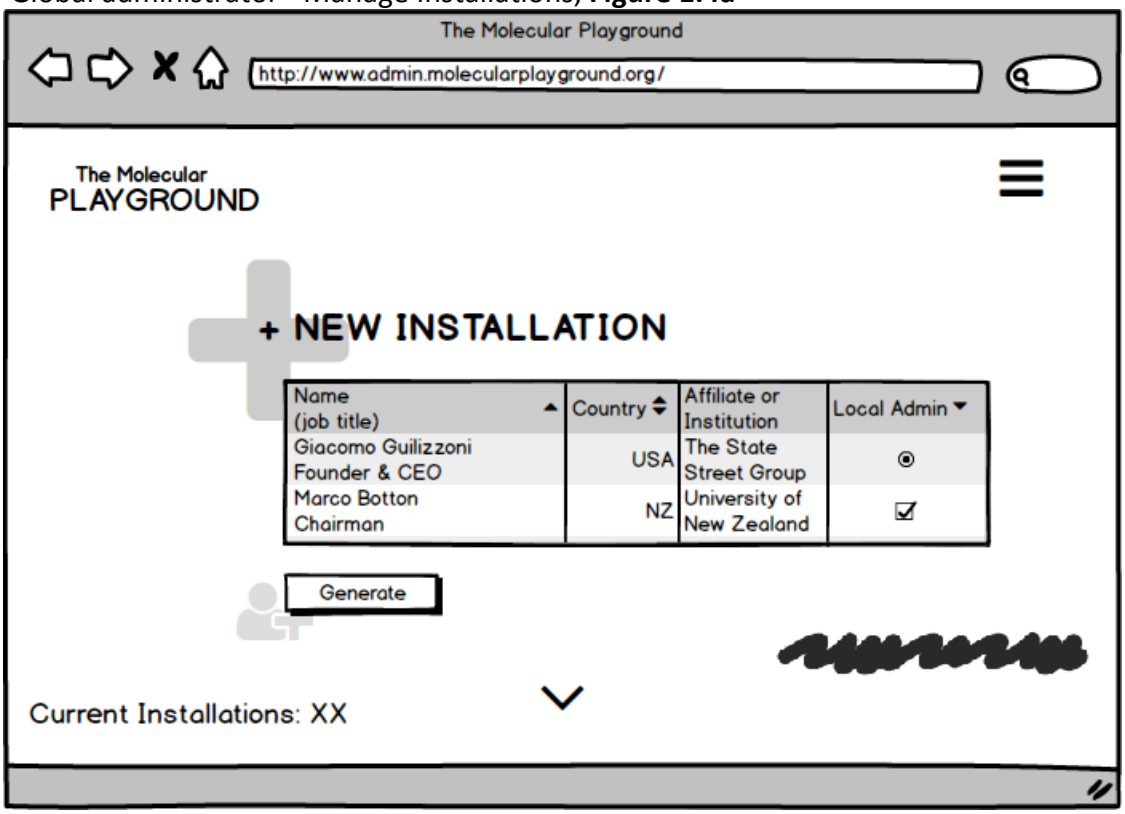
Global administrator - Manage Pending Requests, **Figure 1.3b**



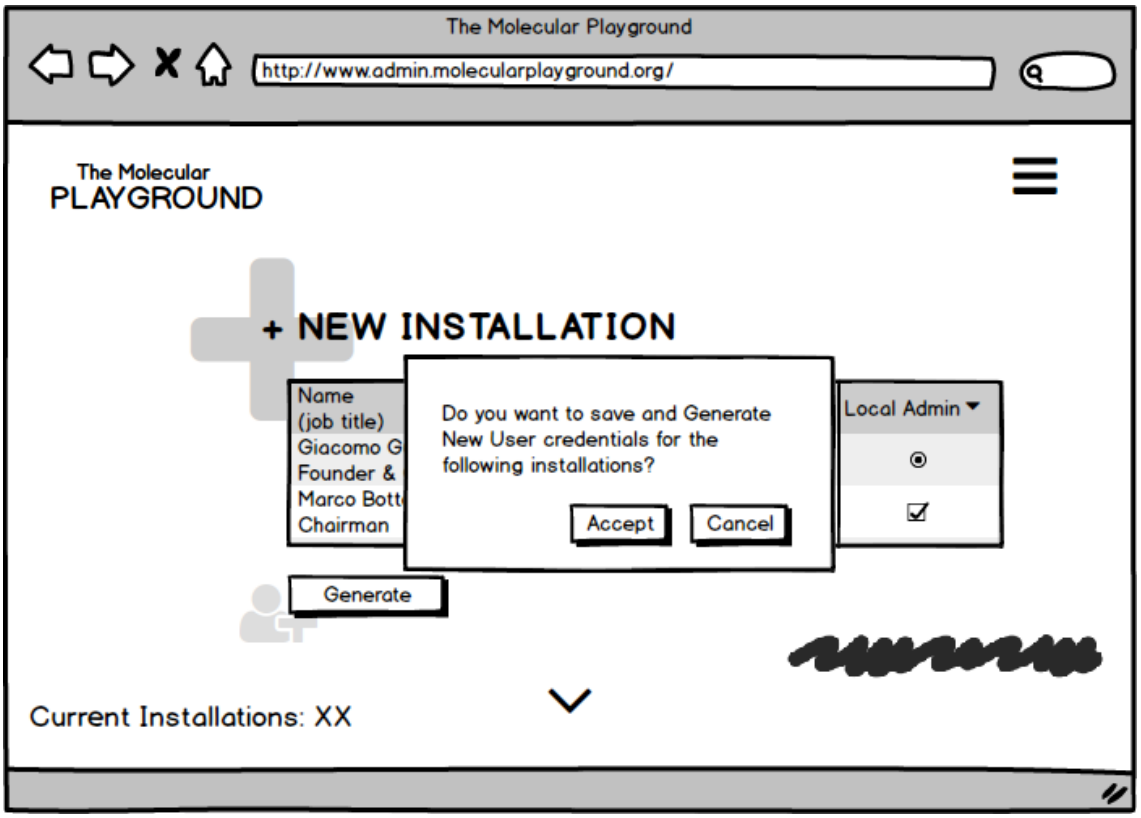
Global administrator - Manage Pending Requests Prompt, **Figure 1.3c**



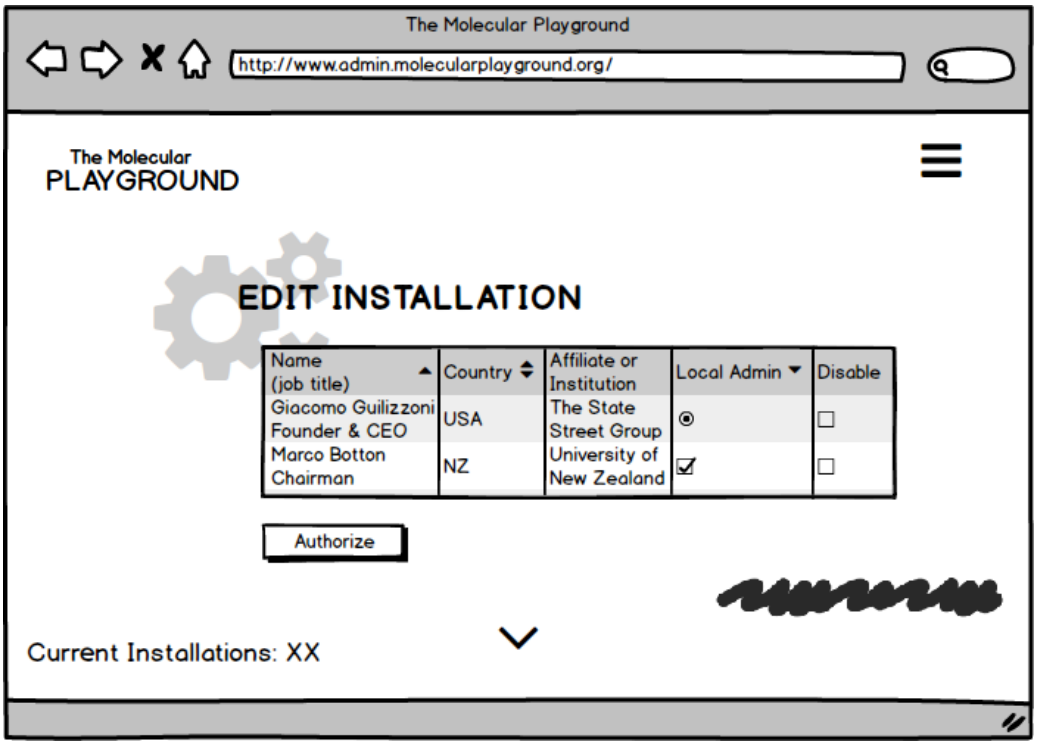
Global administrator - Manage Installations, **Figure 1.4a**



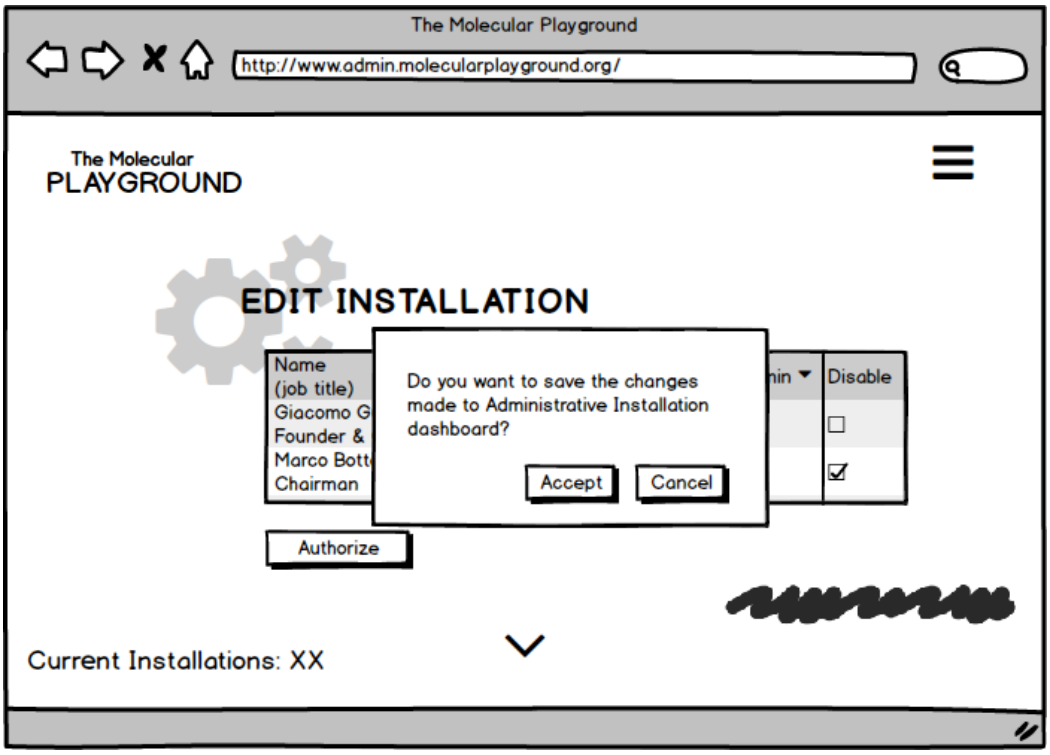
Global administrator - New Installation Page, **Figure 1.4b**



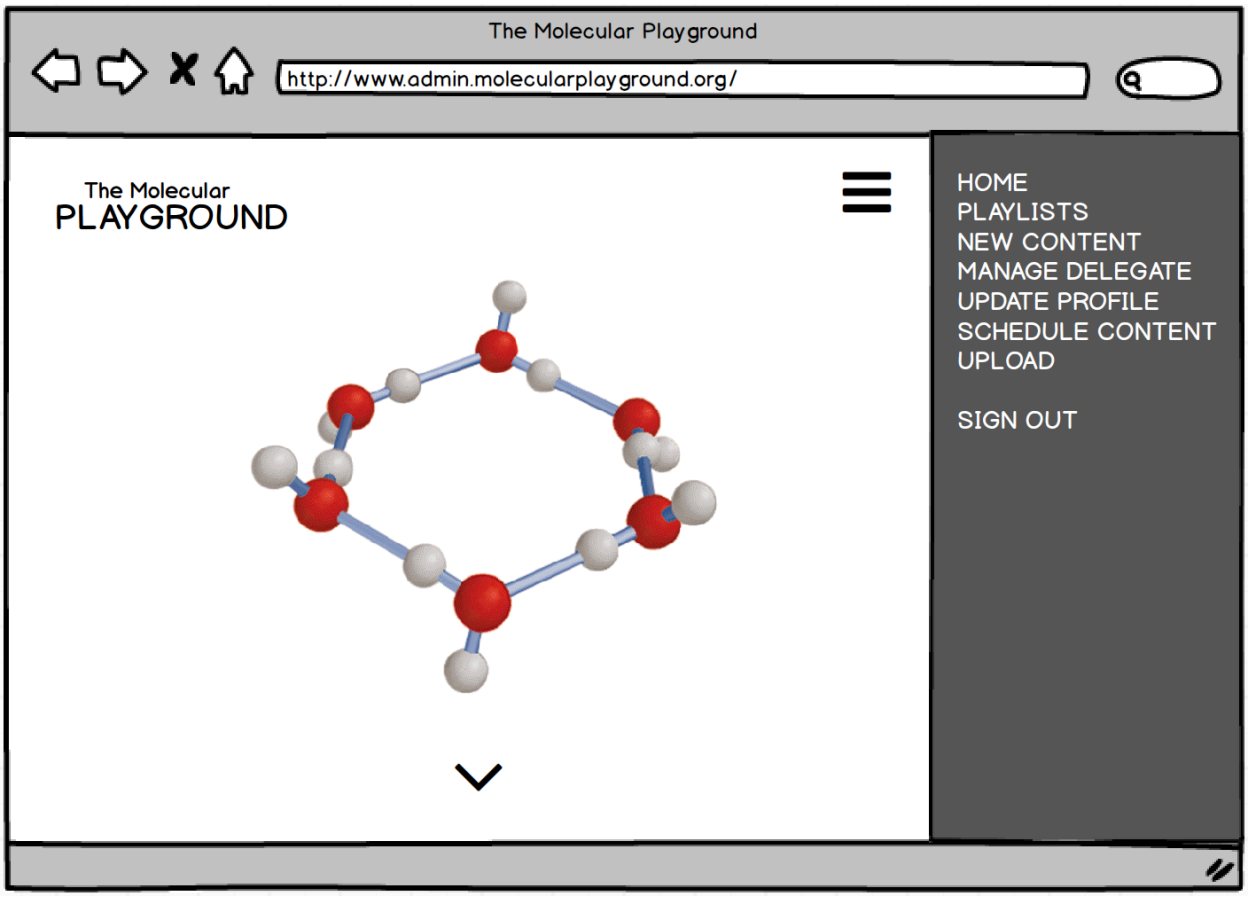
Global administrator - New Installation Prompt, **Figure 1.4c**



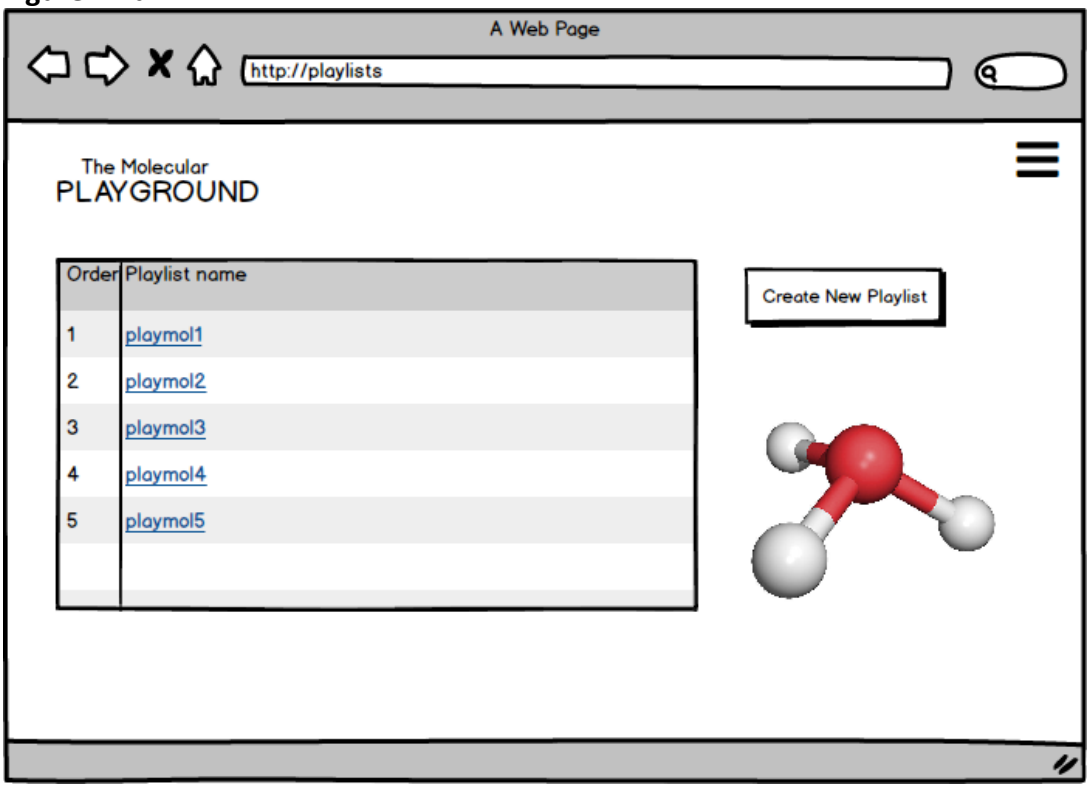
Global administrator - Edit Installation Page, **Figure 1.4d**



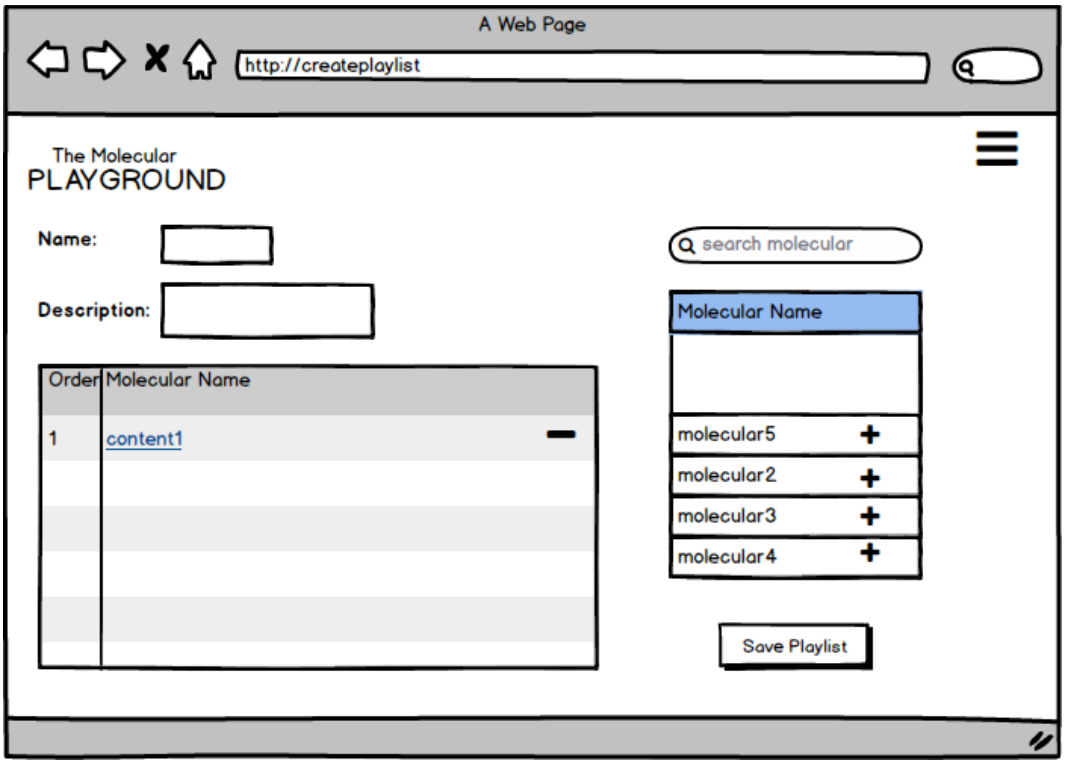
Global administrator - Edit Installation Prompt, **Figure 1.4e**



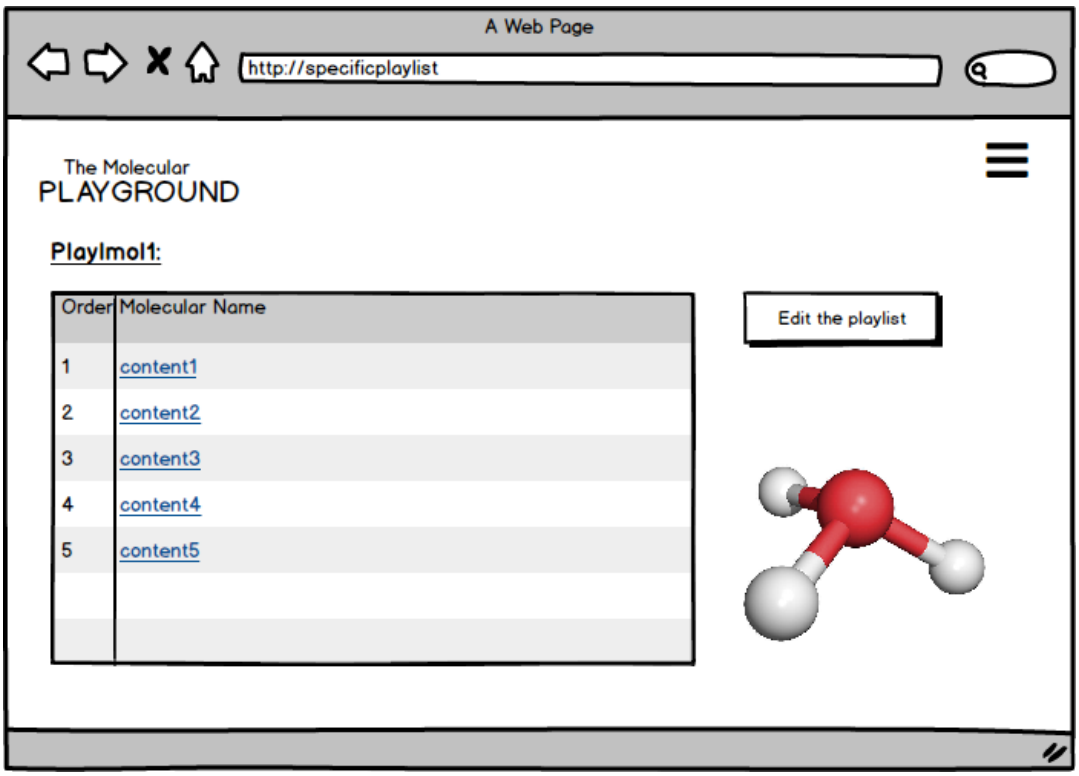
Local administrator (delegates can see all of options except manage delegate)- home page, **Figure 2.1a**



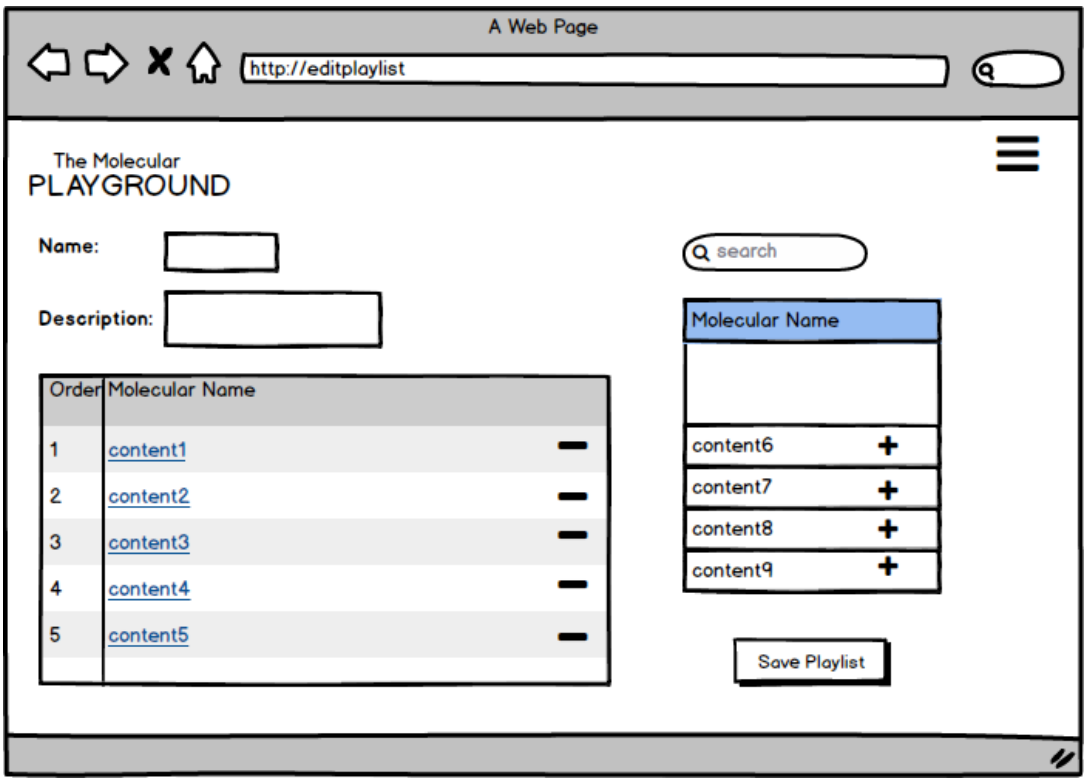
Local administrator/delegates - List of playlists, **Figure 2.1b**



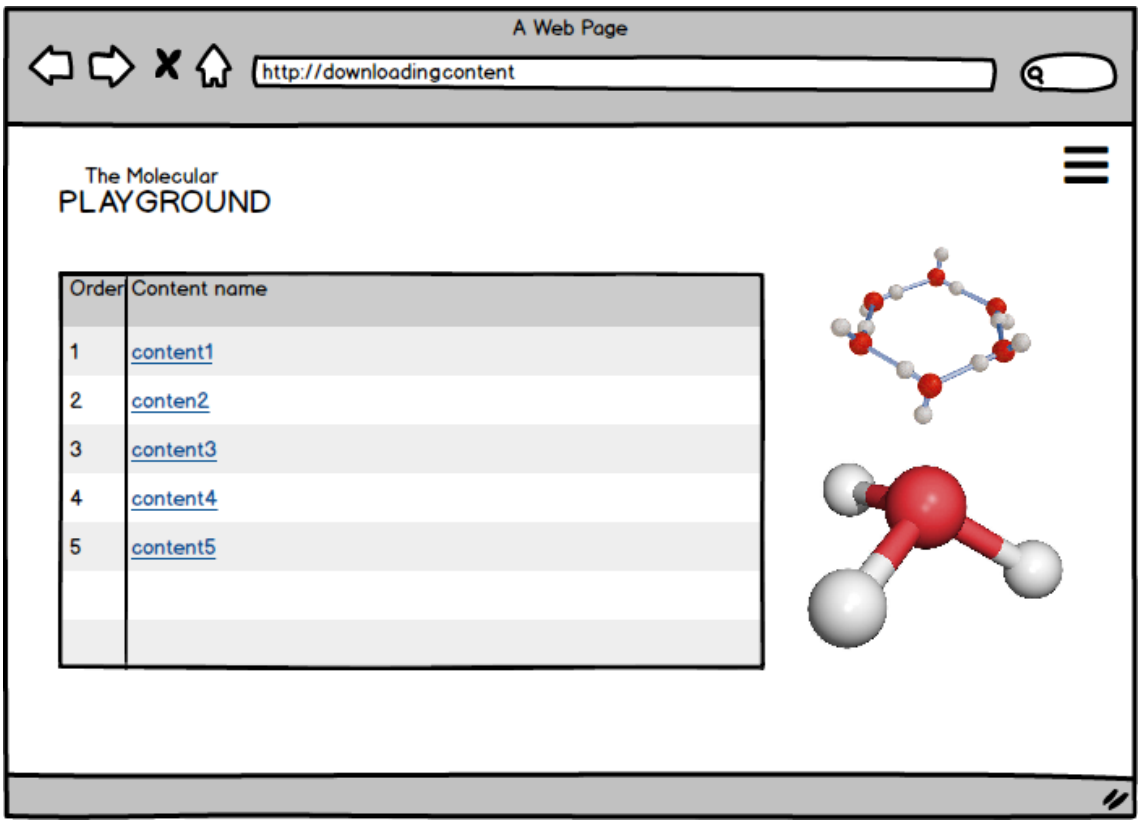
Local administrator/delegate - Create a new playlist, **Figure 2.1c**



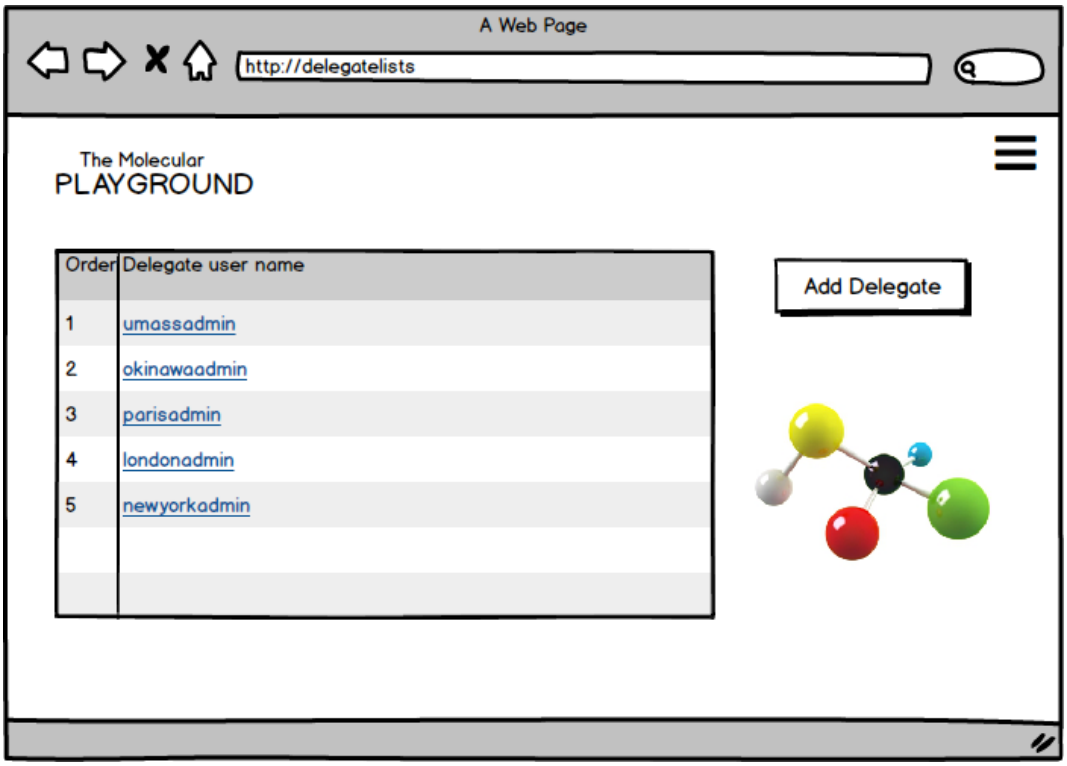
Local administrator/ delegate - A specific playlist, **Figure 2.1d**



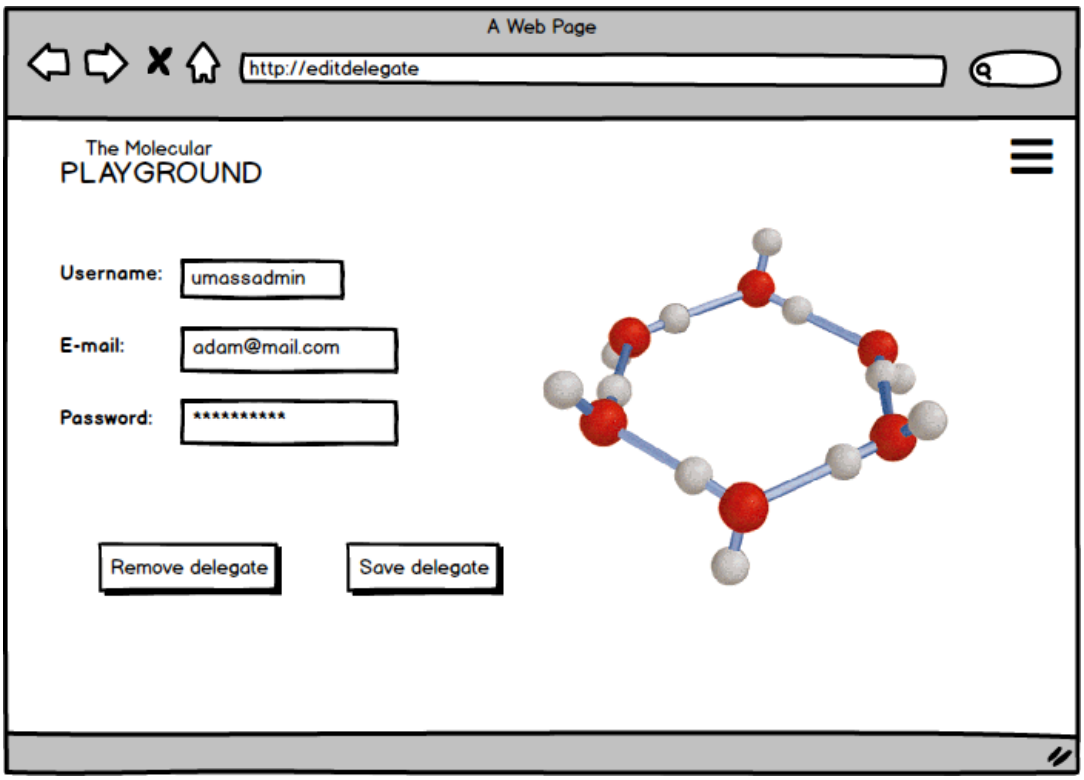
Local administrator/delegate - Edit a specific playlist, **Figure 2.1e**



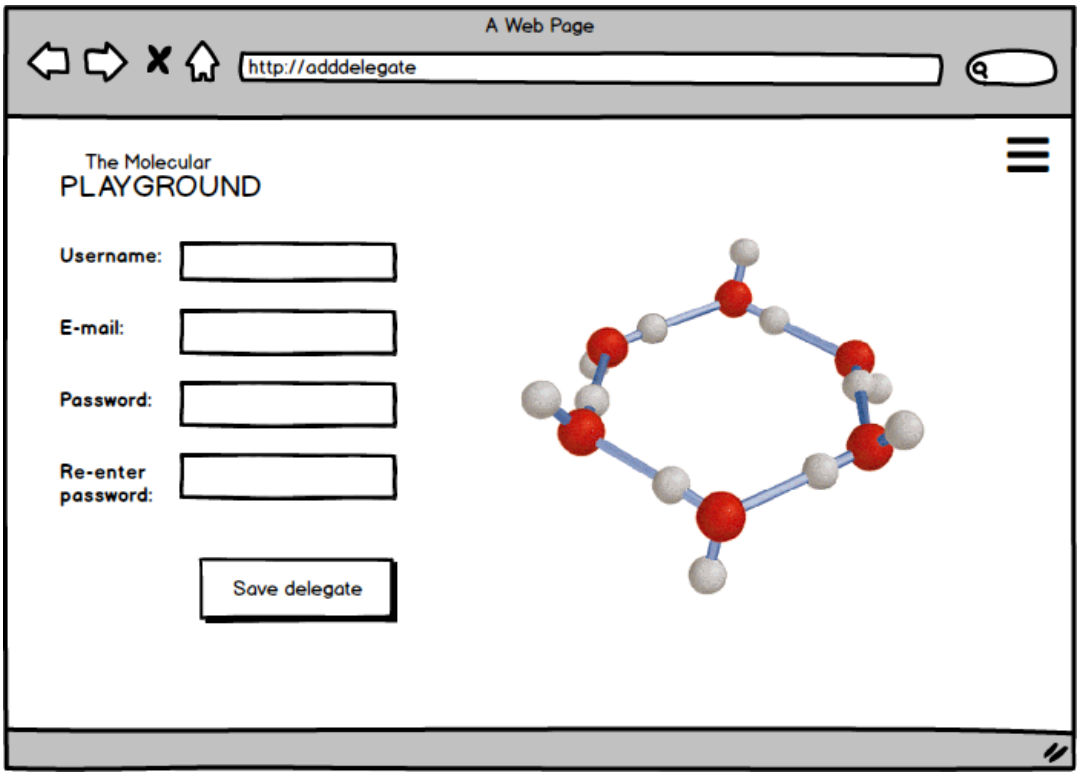
Local administrator/delegate - List of all contents, **Figure 2.2a**



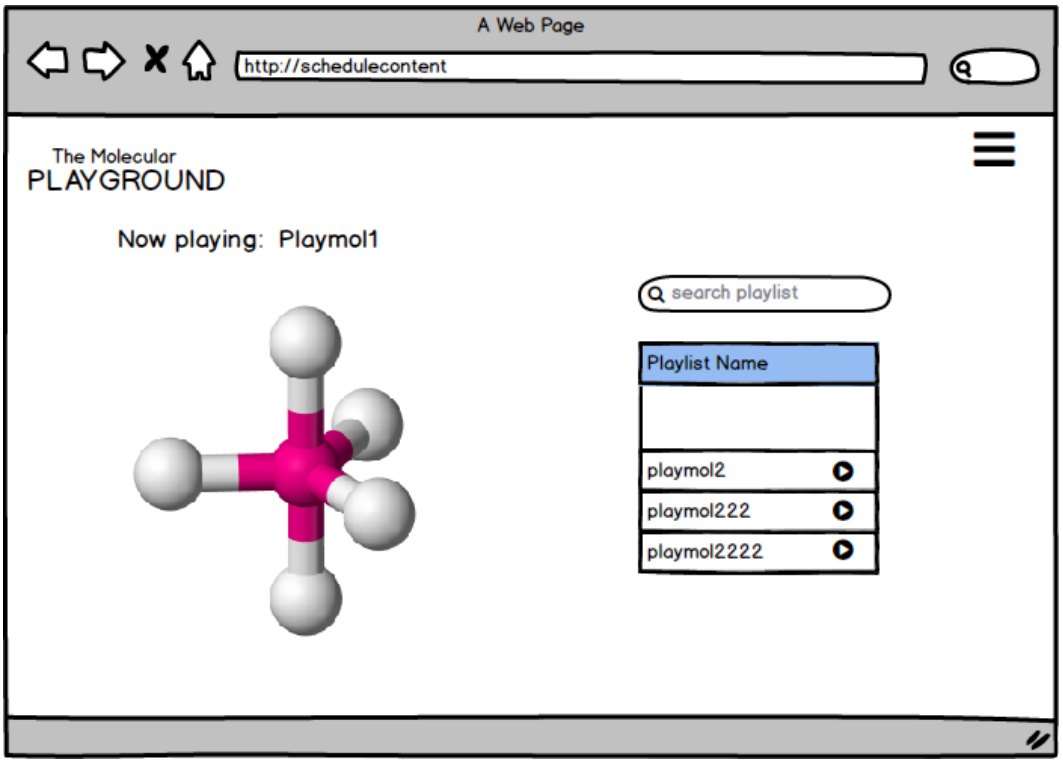
Local administrator - List of delegates page, **Figure 2.3a**



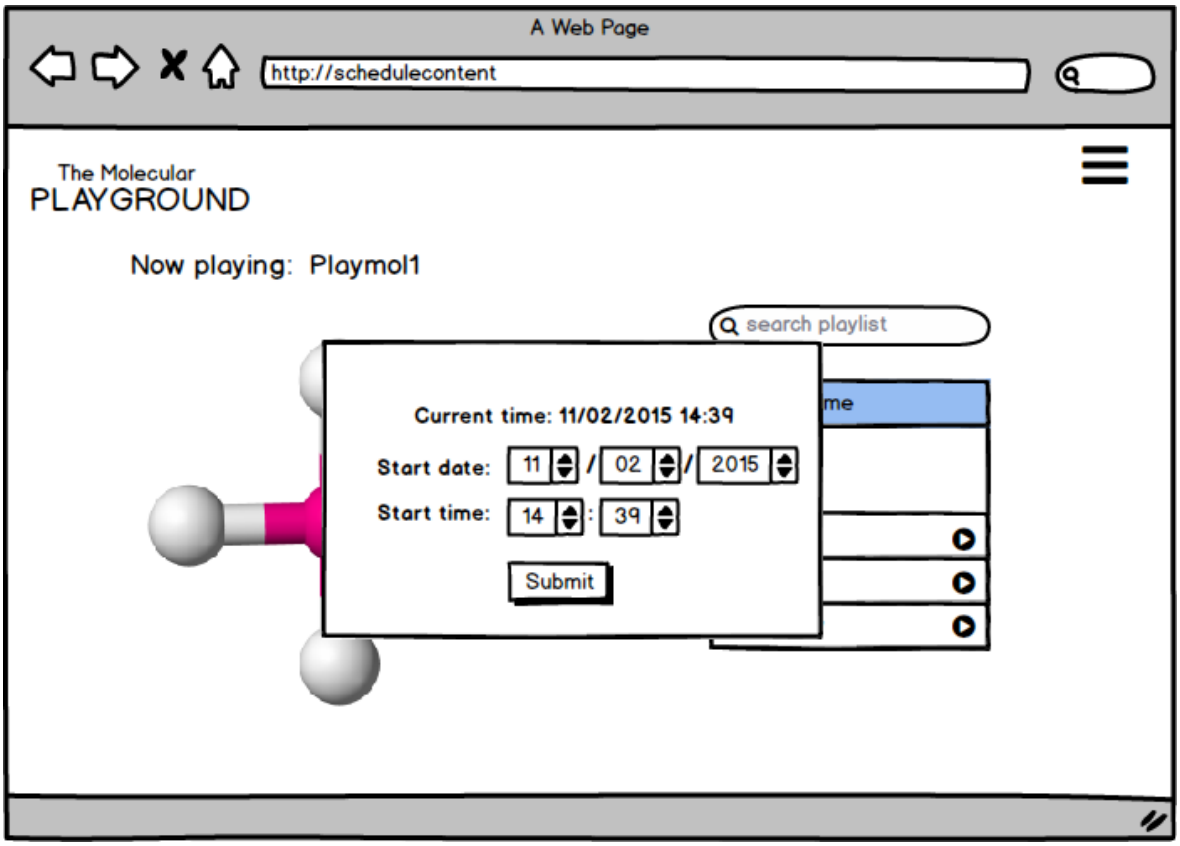
Local administrator - Edit delegate page, **Figure 2.3b**



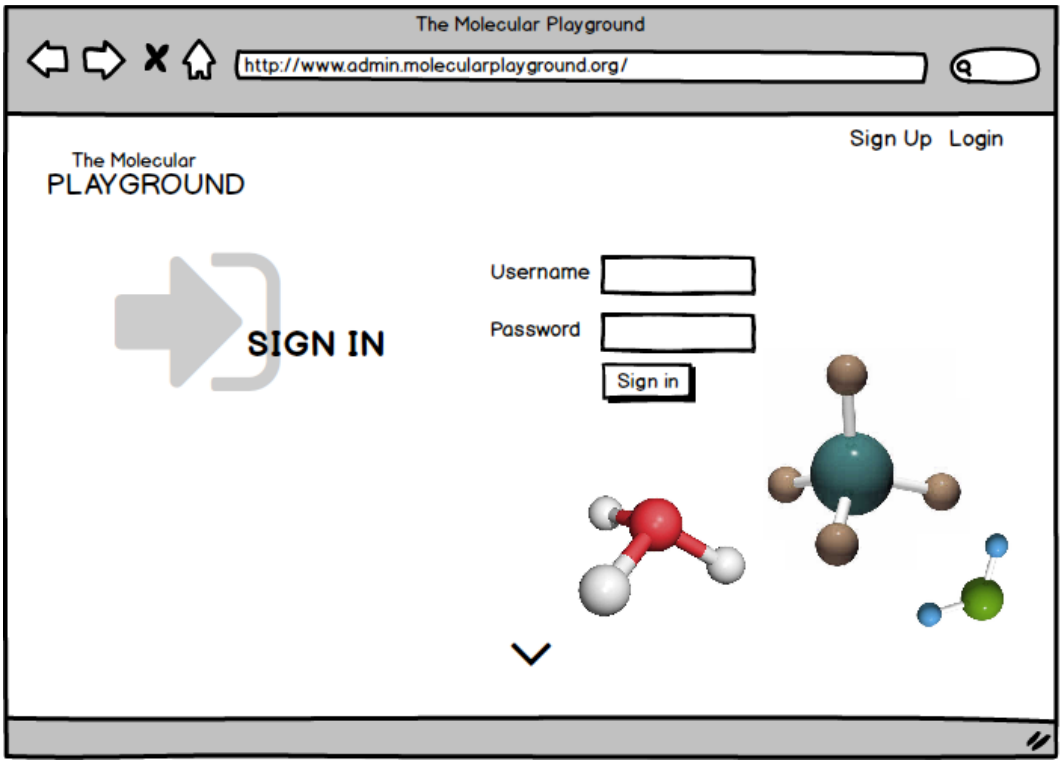
Local administrator - Add delegate page, **Figure 2.3d**



Local administrator/delegate - Schedule content page, **Figure 2.4a**



Local administrator/delegate - Schedule content and time input page, **Figure 2.4b**



All users login page, **Figure 3.1**

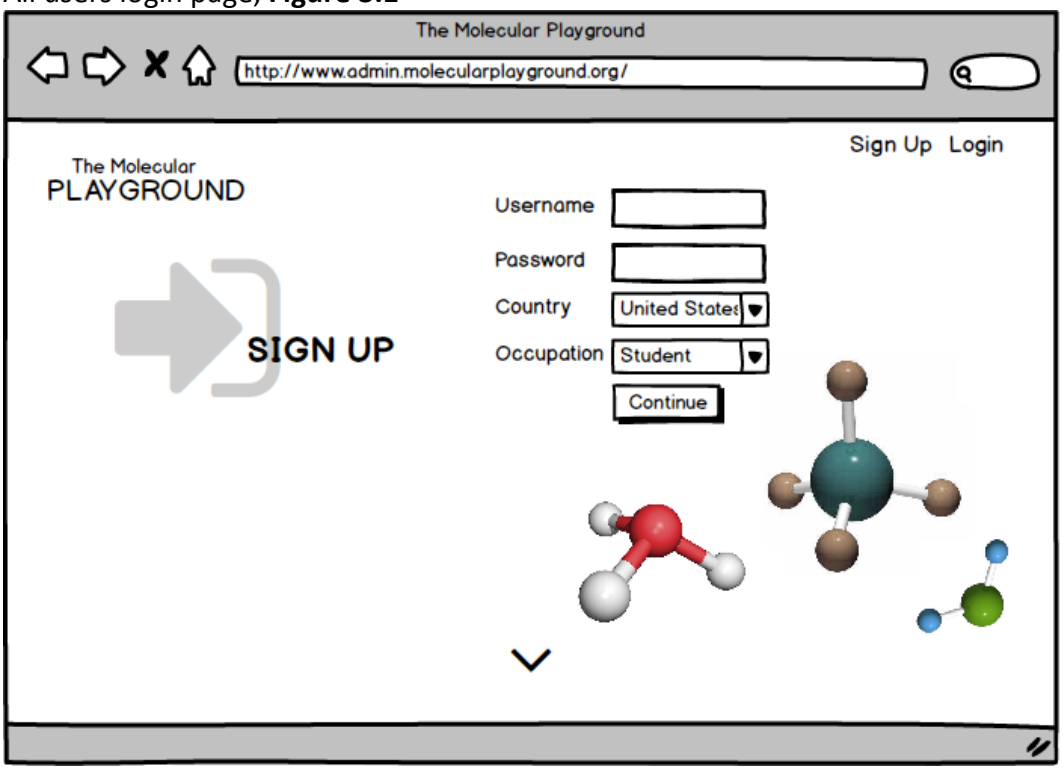


Figure 3.2. Sign-Up for author account

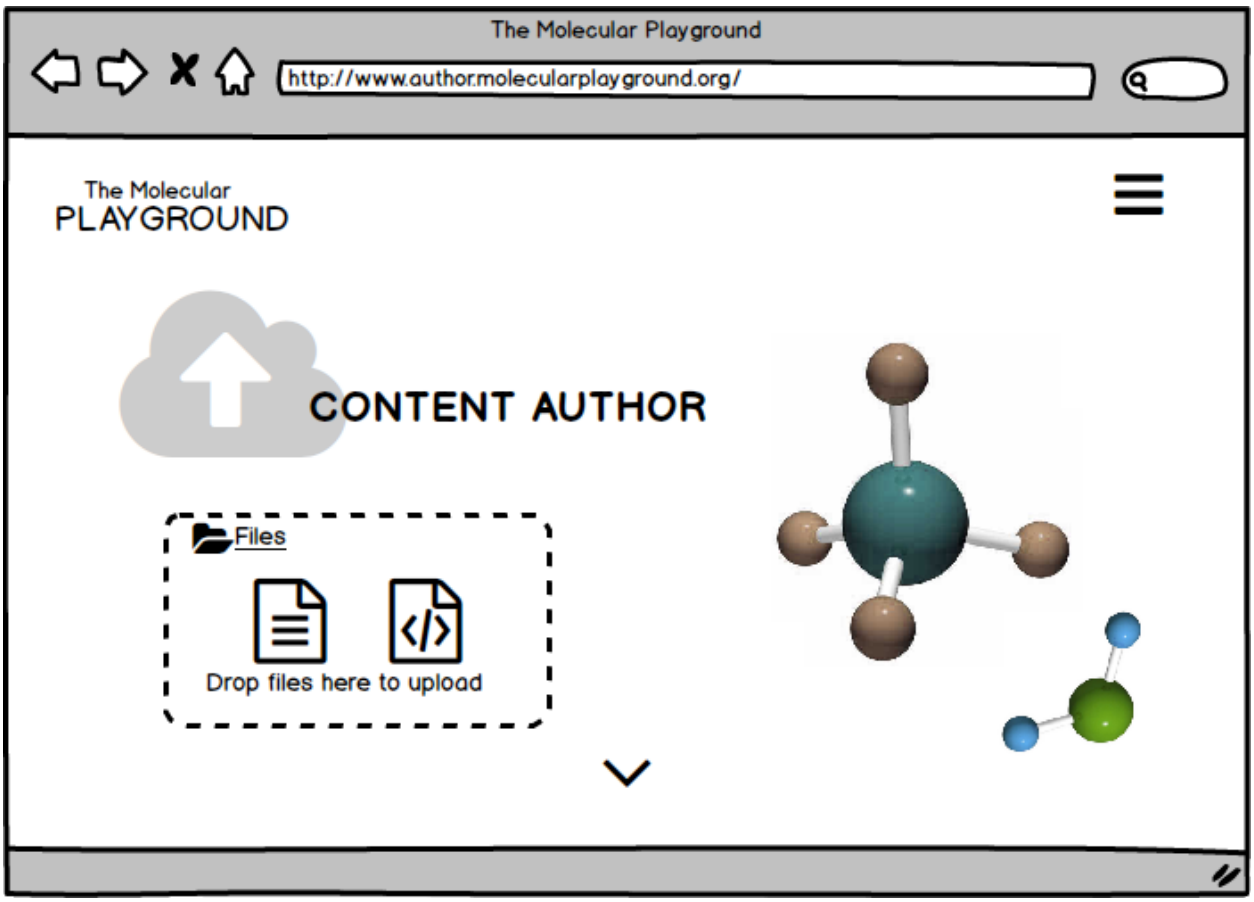


Figure 3.3a. Local administrator, delegate, author upload page

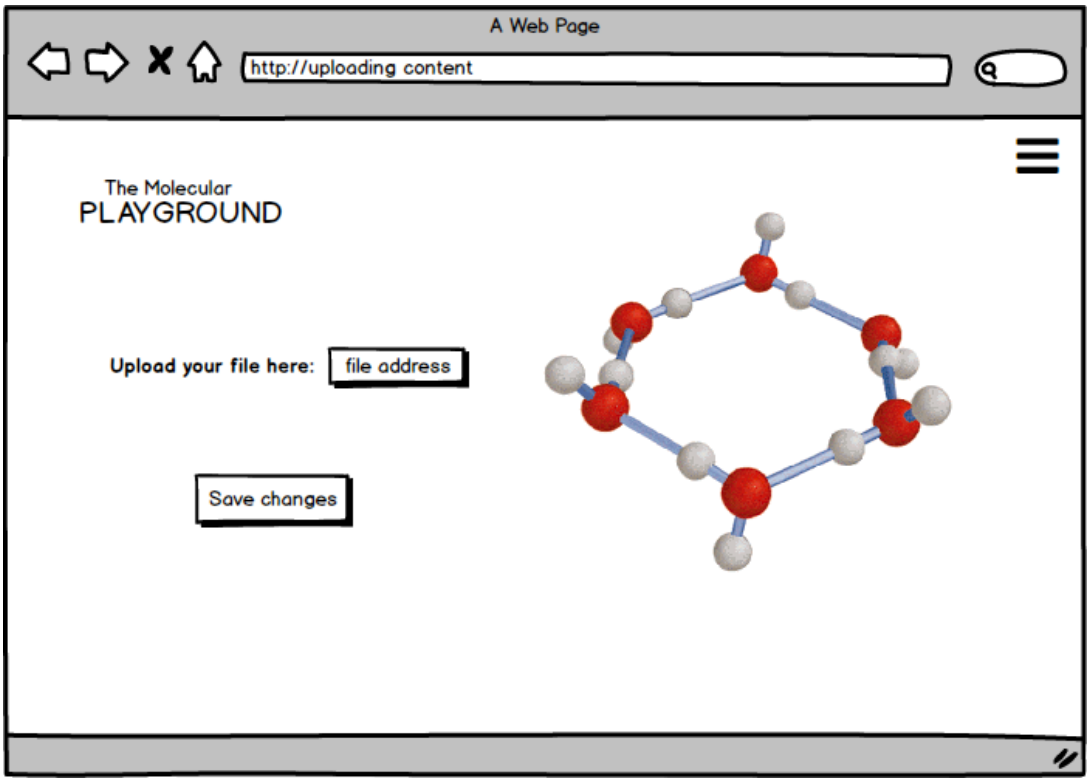


Figure 3.3b. Content Author Upload Page -- uploading (alternate wireframe)

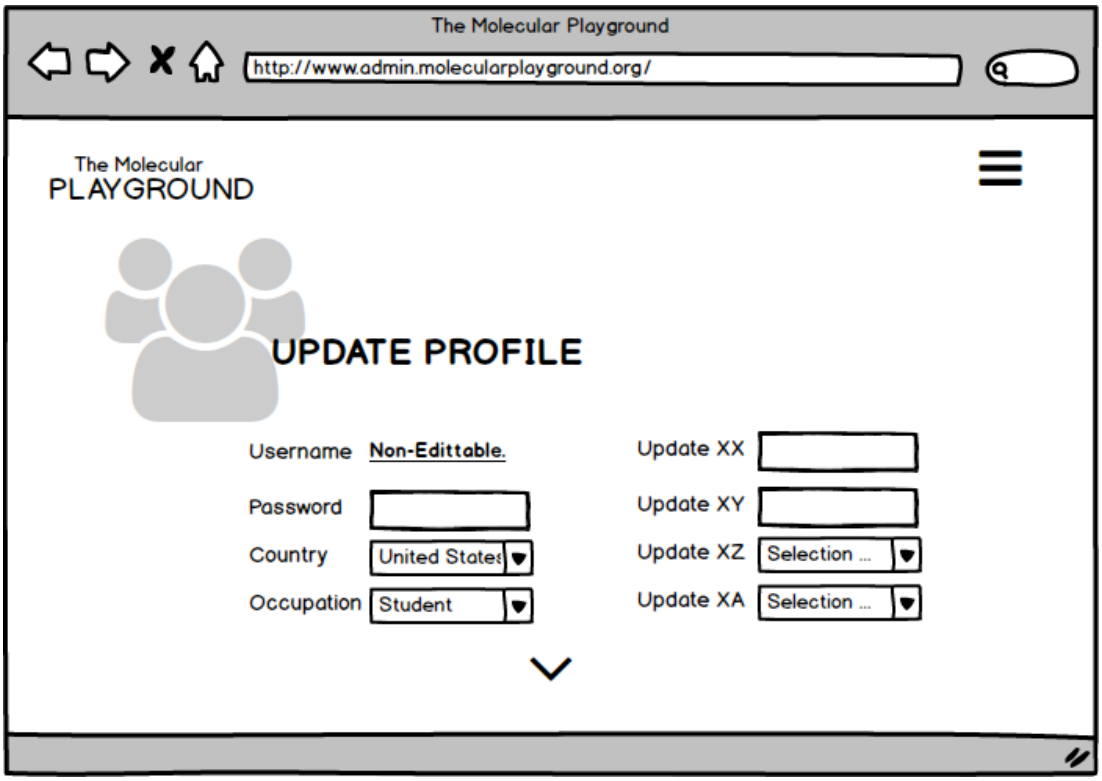


Figure 3.4. Update Profile Page for global administrators, authors, local administrators, and delegates