Team: Paul Boschert, Teresa Creech

Title: Tardis-box

Project Summary:

Tardis-box is a no-charge media distribution system. It will allow a guest to view and sort the available media and, if desired, create a member account using a valid email address. The member may delete this account, if all borrowed media has been checked in.

A member may view and sort the media with the option to check-out an item. If the requested media is unavailable, the request can be waitlisted to be checked out to the user as soon as it is available. The member may check-in media.

An administrator may view and sort the media. Only an admin can add and delete media. An administrator can delete a member account, if necessary and can handle overdue items.

Project Requirements:

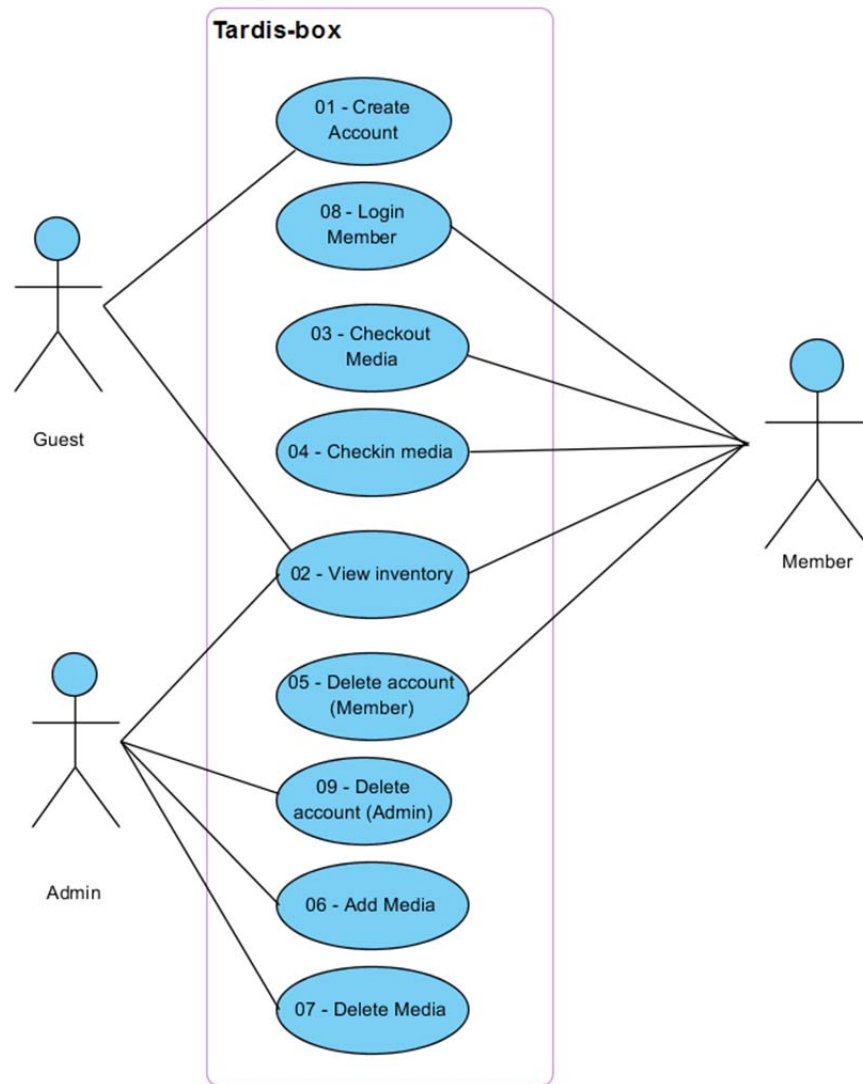| Business Requirements				
ID	**Requirement**	**Topic Area**	**User**	**Priority**
BR-01	Member usernames must be a valid e-mail address.	Accounts	Member	Medium
BR-02	Only admin can add/delete media items.	Database	Admin	Medium
BR-03	Guests can view available media without having to create an account or login.	Database	Guest	Critical

User Requirements				
ID	**Requirement**	**Topic Area**	**User**	**Priority**
UR-01	Guest can request and create their own account	Accounts	Guest	Critical
(Help Page)				

UR-02	Any user can view media list	Database	All	Critical
UR-03	A member can remove their account if certain conditions are met (e.g. no overdue items).	Accounts	Member	Medium
UR-04	A member can make a request to checkout media.	Database	Member	Critical
UR-05	A member can check-in media.	Database	Member	Critical
UR-06	An admin can delete an account	Accounts	Admin	Medium
UR-07	An admin can handle overdue items	Accounts	Admin	Medium

Functional Requirements				
ID	**Requirement**	**Topic Area**	**User**	**Priority**
FR-01	There must be a display of the media that shows the title and an associated picture.	Database	All	Critical
FR-02	There must be a way to view details of a particular media item (besides just a picture and title) that shows a short description and possibly other details e.g. IMDB data).	Database	All	Medium
FR-03	There must be a way to sort the media for display to the user.	Database	All	Medium
FR-04	There must be a way to put media on a waitlist, if necessary.	Accounts	Admin	Medium
FR-05	There must be a way to remove item from waitlist (and check it out to a requestor)		Admin	Medium

Non-Functional Requirements				
ID	**Requirement**	**Topic Area**	**User**	**Priority**
NR-01	Time to cycle from one page media titles to the next must be reasonable.	User Interface	All	Medium
NR-02	Member credentials must be encrypted.	Accounts	Member	Low
NR-03	Web pages need to be clearly worded and easy to navigate.	User Interface	All	Medium

Users and Tasks:

Use Case ID:	UC-01	
Use Case Name:	Create Account	
Description:	Allow a guest to create a member account, which will allow member access to the database of items to check out.	
Actor:	Guest	
Pre-conditions:	Member database must exist	
Post-conditions:	Guest becomes a Member	
Frequency of Use:	Unknown, potentially 100% of guests	
Flow of Events:	**Actor Action**	**System Response**
	1. Guest visits homepage and clicks on "create account" button	Accesses the create account page
	2. Guest enters first and last names, a valid email-id, enters a password, then presses "ok"	System verifies that the email is valid, creates entry in the member database containing this information. System issues an "account created" message
Variations:	Guest enters an invalid email-id. System issues error message then clears the create account page. Guest omits email-id and/or password – error message issued screen cleared.	
Notes & Issues:		
Developer Notes:		

Use Case ID:	UC-02	
Use Case Name:	View Inventory	
Description:	Allow a guest or member to access the item database. This will allow them to scroll through and sort (by item number, name, or description)	
Actor:	All	
Pre-conditions:	Database exists with ability to sort and scroll, Guest/Member has accessed the View Media page	
Post-conditions:	Database is scrolled and/or sorted	
Frequency of Use:	Arbitrary, depends on guest/member needs	
Flow of Events:	**Actor Action**	**System Response**
	1. Uses scroll bar	Moves list up or down, depending on the direction user is moving scroll bar
	2. Sort by clicking one of the column headers (once for ascending order, twice for descending order)	Sorts the list using the selected column, either in ascending or descending order as indicated by the number of clicks.

	3.	Double click item	Displays additional information
Variations:			
Notes & Issues:	The check-in/out buttons will be grayed out when page is accessed from the home page instead of the sign in page.		
Developer Notes:			

Use Case ID:	**UC-03**		
Use Case Name:	Check-out Media		
Description:	Member may borrow an item if it is available (not already checked-out).		
Actor:	Member		
Pre-conditions:	Member is signed in and has accessed the View Media page		
Post-conditions:	Item is either checked-out to the member or has been placed on a waitlist for the member.		
Frequency of Use:	Arbitrary, estimated at least once per member login.		
Flow of Events:		**Actor Action**	**System Response**
	1.	Enters an item number into the check-out field, Clicks OK button	Checks to see if the item is available. If it is, the item is checked-out to the member. If not available, system issues message asking member if he wishes to add the item to the waitlist.
	2.	Selects yes	Item is added to waitlist and system issues confirmation message.
	3.	Selects no	System removes message
Variations:	Issue error message if the item number is invalid.		
Notes & Issues:			
Developer Notes:			

Use Case ID:	**UC-04**
Use Case Name:	Checkin Media
Description:	Member returns an item which had been borrowed
Actor:	Member
Pre-conditions:	The item must already be checked-out, member is logged in and has accessed the view items page
Post-conditions:	The item is available for check-out
Frequency of Use:	Done for up to 100% of items borrowed, estimate many times daily

Flow of Events:		Actor Action	System Response
	1.	Enters item number in the check-in field, presses OK	System makes item available for check-out, updates item list, and removes it from the member's account. Issues a "check-in successful" message.
	2.		If item is on waitlist, system checks it out to the next user who requested the item.
	3.		If item is not on waitlist, system makes item available for general checkout
Variations:	Members enters an invalid item number - either already available in the database or the item does not exist. An appropriate error message is displayed for each error and the check-in field is cleared.		
Notes & Issues:			
Developer Notes:			

Use Case ID:	UC-05		
Use Case Name:	Delete Account (Member)		
Description:	Member may delete account as long as all items have been checked-in		
Actor:	Member		
Pre-conditions:	Must be on login page		
Post-conditions:	Account no longer exists		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1.	Enters email-id and password	Verifies information
	2.	Clicks on Delete button	Verifies that no items are checked-out, removes account and issues confirmation email
Variations:	Email-id and/or Password invalid – system issues error message and clears screen Items checked out – system issues message instructing Member to return items before account can be deleted		
Notes & Issues:			
Developer Notes:			

Use Case ID:	UC-06
Use Case Name:	Add media
Description:	A new media item can be added to the system.

Actor:	Admin
Pre-conditions:	The media item must not already exist.
Post-conditions:	The new media item was added and is available for checkout.
Frequency of Use:	Arbitrary, anytime an admin needs to add media. Est: weekly

Flow of Events:		**Actor Action**	**System Response**
	1.	Log in to Tardis-box	User credentials are verified
	2.	Click on administrator panel	Administrative options are displayed
	3.	Click on media sub-panel	A list of media actions is displayed
	4.	Click on add media	The media entry form is displayed
	5.	Enter media details; click submit	The new media item is added

Variations:	
Notes & Issues:	
Developer Notes:	

Use Case ID:	**UC-07**
Use Case Name:	Delete media
Description:	A media item needs to be removed from the system.
Actor:	Admin
Pre-conditions:	The media item must exist.
Post-conditions:	The media item deleted no longer exists in the system.
Frequency of Use:	Arbitrary, anytime an admin needs to delete media. Est: weekly

Flow of Events:		**Actor Action**	**System Response**
	1.	Log into Tardis-box	User credentials are verified
	2.	Click on administrator panel	Administrative options are displayed
	3.	Click on media sub-panel	A list of media actions is displayed
	4.	Click on delete media	A list of available media items is displayed
	5.	Select the media item; click delete	The media item is deleted

Variations:	
Notes & Issues:	
Developer Notes:	

Use Case ID:	**UC-08**

Use Case Name:	Login Member	
Description:	A member may login to account	
Actor:	Member	
Pre-conditions:	Member is on login page	
Post-conditions:	Member is logged in	
Frequency of Use:		
Flow of Events:	**Actor Action**	**System Response**
	1. Enter email-id and password, clicks on Login button	Verifies information, if account is an admin account, the system will go to the admin page. Members are logged in and sent to view items page
	2.	If waitlisted item is now available, system displays message indicating that the item is available for check-out and will be deleted from the waitlist (checkout is in different UC table)
Variations:	Email-id and/or Password invalid – system issues error message and clears screen	
Notes & Issues:		
Developer Notes:		

Use Case ID:	**UC-09**	
Use Case Name:	Delete account (Admin)	
Description:	Administrators can remove user accounts	
Actor:	Admin	
Pre-conditions:	The user that needs to be deleted exists.	
Post-conditions:	The deleted user no longer exists.	
Frequency of Use:	Arbitrary, anytime a user needs to be removed by an administrator. Est: monthly.	
Flow of Events:	**Actor Action**	**System Response**
	1. Log in to Tardis-box	User credentials are verified
	2. Click on administrator panel	Administrative options are displayed
	3. Click on user sub-panel	A list of users is displayed
	4. Select a user	The user is highlighted
	5. Click the remove icon	The user is removed

Variations:	
Notes & Issues:	
Developer Notes:	

Activity Diagram:

Tardis-box Check-in Media (UC #04)

Member

Enters item number and clicks OK

System

Verify item number

Valid Item Num? — YES → Checked out by member? — YES → Remove item from Member's borrowed list → Issue Check-in Successful message → Item on Waitlist? — NO → (end)

NO → Issue error message

NO → Issue error message

Item on Waitlist? — YES → Remove item from waitlist and check-out to member
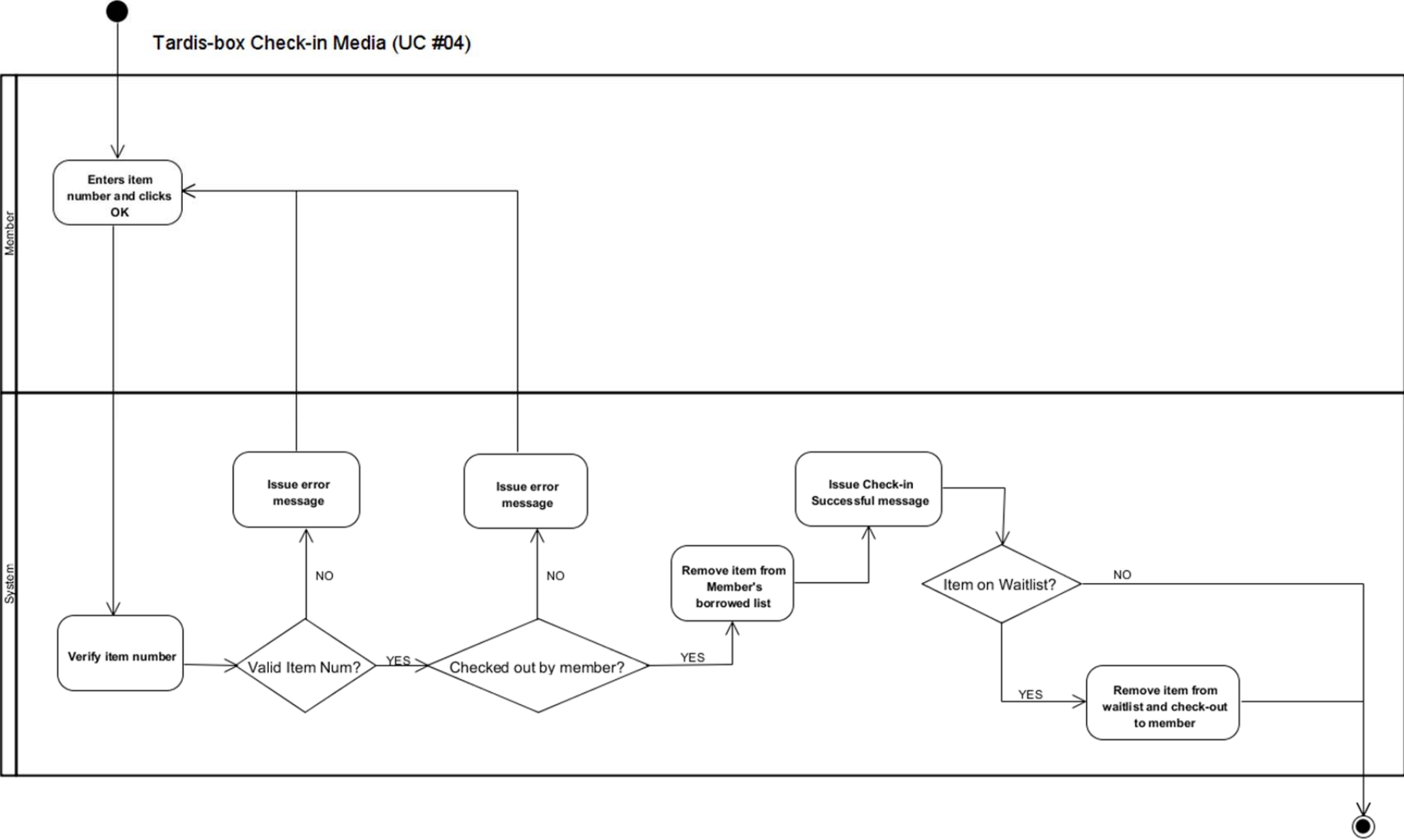
Data Storage:
The data for the media distribution system will be stored in a relational database. Preliminarily we're planning to use MySQL. But we're designing our system to be agnostic to the specific relational database server.
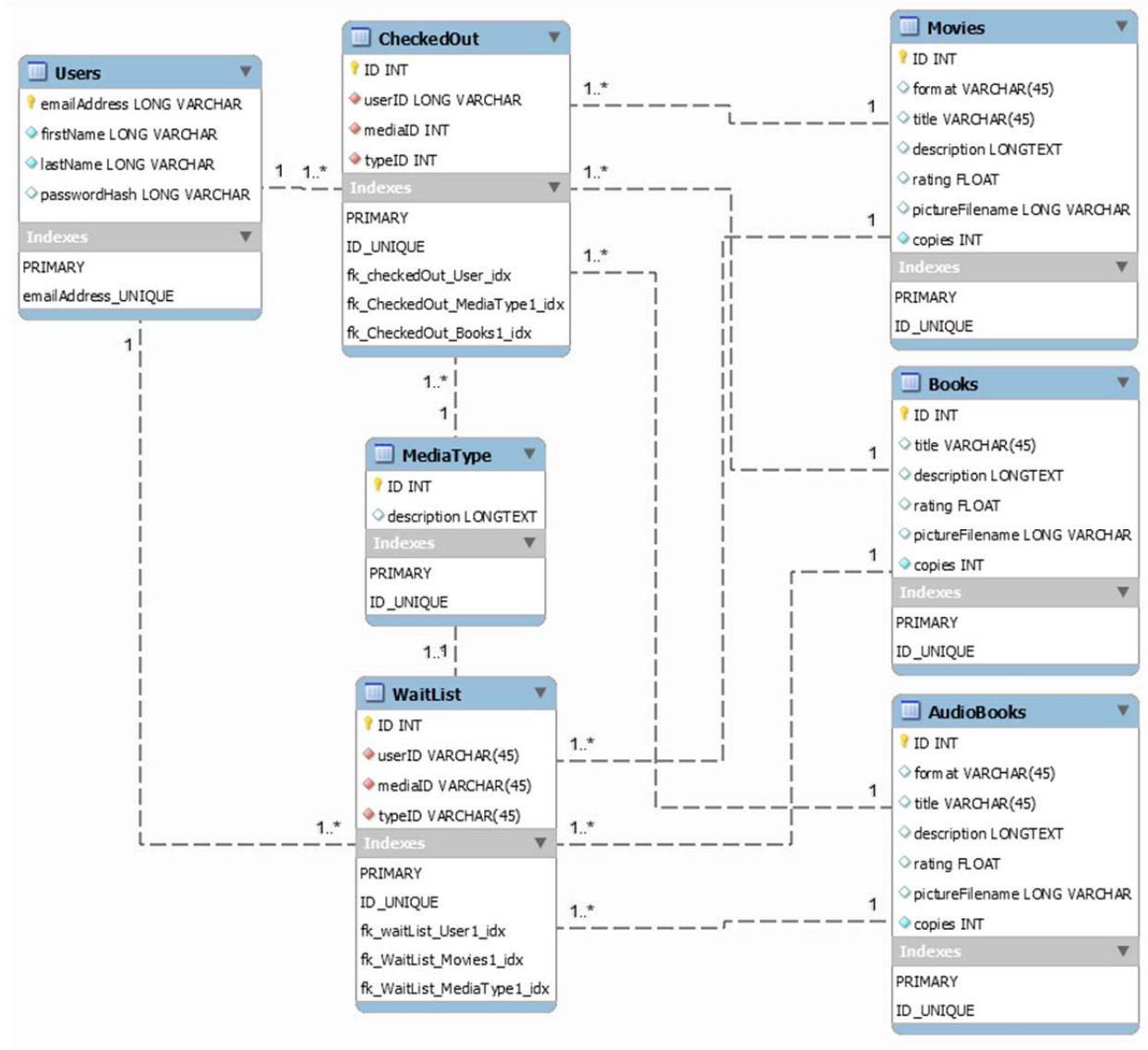
UI Mockups:
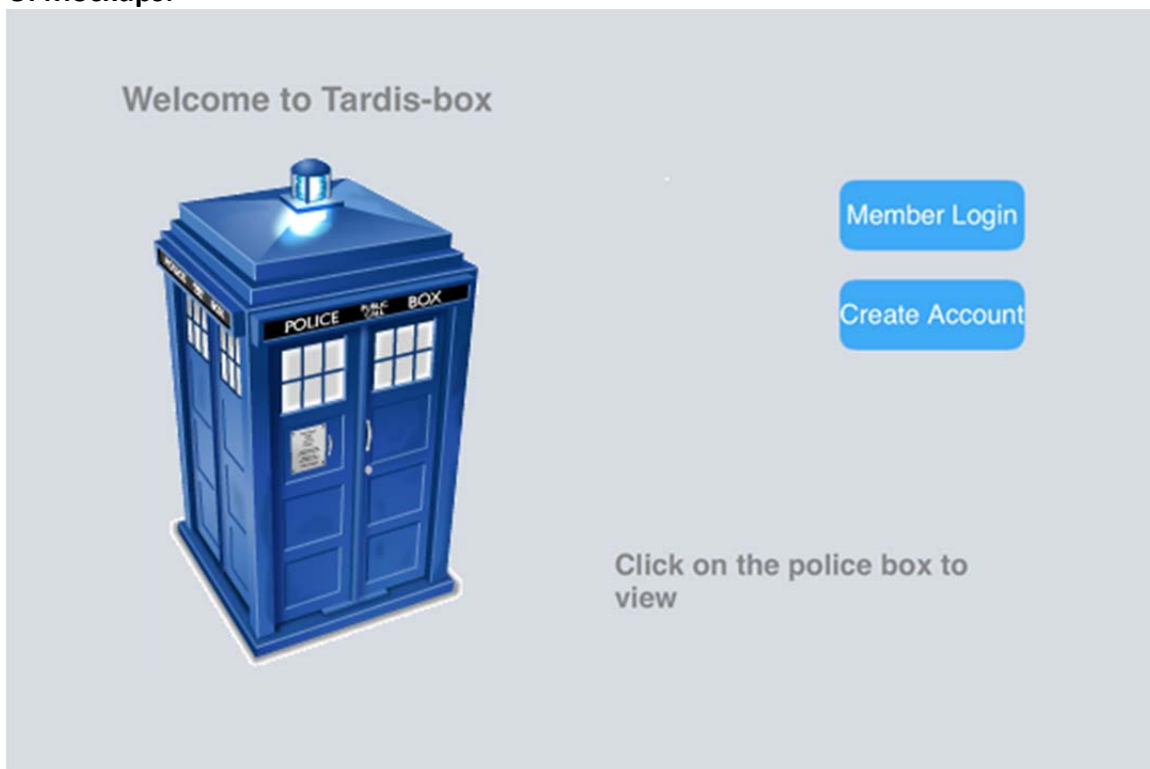
Figure 1: Home Page
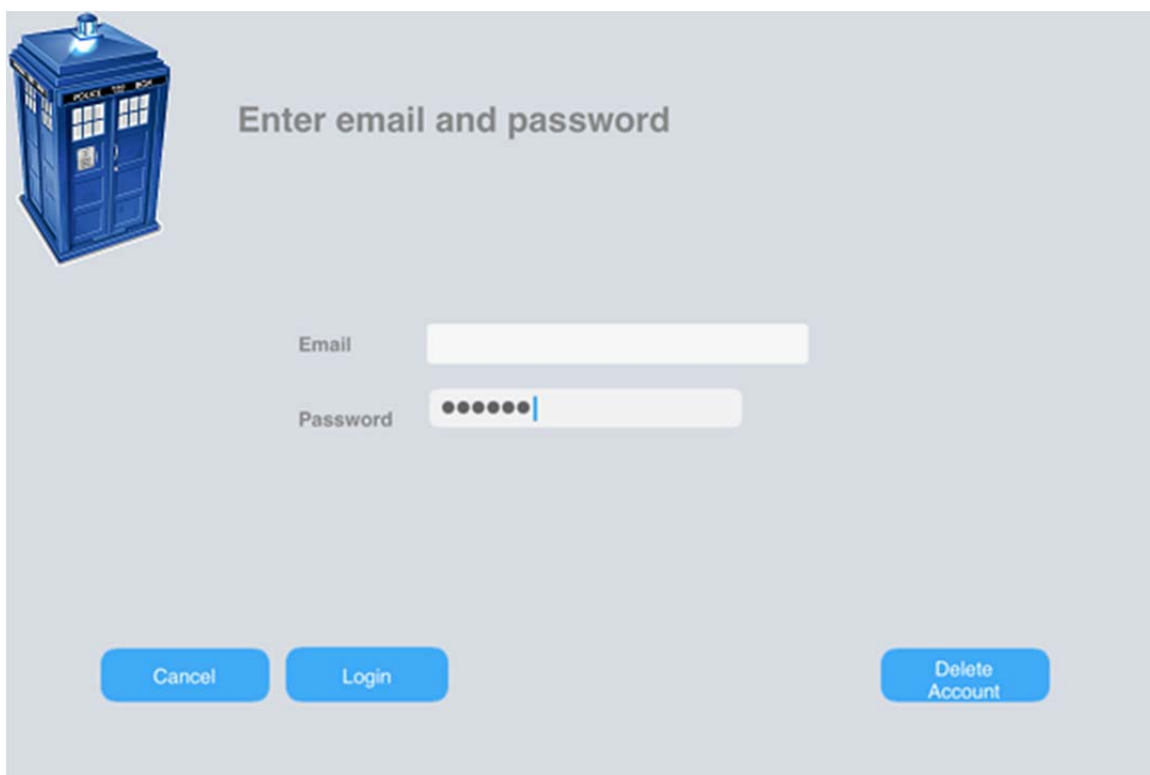
Figure 2: Member Login Page
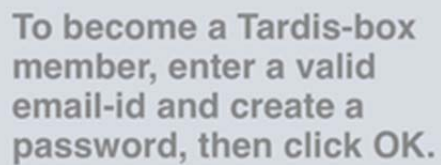
Figure 3: Create Account Page

Figure 4: View Media Page

Tardis-box Admin Page

View Items

This button goes to the view items page allowing the admin to scroll through and sort the item database

Add Items

Allows an admin to add one or more item to the database.

Delete Items

Deleting an item will remove it from the inventory database, the waitlist, and update a user account if the item shows as being checked out.

Delete Account

This action will delete an account from the system. It will release any item currently loaned to the account and clear any waitlisted items. NOTE: If the item is missing or damaged, it needs to be removed from the database using the Delete Items function.

Figure 5: Admin Panel Page

Tardis-box Admin: Add Media

Media entry form to be inserted here

Add Item

Add Item and Exit

Figure 6: Add Media Page
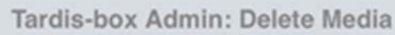
Tardis-box Admin: Delete Media

Item #	Title	Description
DVD01	Example Title	Example description

Scroll through the Tardis-box inventory list to find items. Can also sort by clicking names of columns (need better words, I'm just filling up space).

Click on the item to be deleted and click OK.

This tool does not have sort arrows - that will go on the column names at the top of the table. Pretend they are there.

OK

Figure 7: Delete Media Page

Tardis-box Admin: Delete Member

Member Name	Email	Password
Teresa Creech	TECR7733@colorado.edu	badpassw0rd

Scroll through the Tardis-box member list to find items. Can also sort by clicking names of columns (need better words, I'm just filling up space).

Click on the member to be deleted and click OK.

This tool does not have sort arrows - that will go on the column names at the top of the table. Pretend they are there.

OK

Figure 8: Delete Member Page

User Interactions:
TODO

Class Diagram:

Movie
- -format : Format
- -imdbId : String
- +getFormat() : Format
- +setFormat(format : Format) : void
- +getImdbId() : String
- +setImdbId(imdbId : String) : void

Book
- -ISBN : String
- -author : String
- +getAuthor()
- +setAuthor(author) : void
- +getISBN() : String
- +setISBN(ISBN : String) : void
- +getAuthor() : String
- +setAuthor(author : String) : void

Media
- -title : String
- -rating : float
- -pictureFilename : String
- -description : String
- -copies : Integer
- -id : String
- +getTitle() : String
- +setTitle(title : String) : void
- +getRating() : float
- +setRating(rating : float) : void
- +getPictureFilename() : String
- +setPictureFilename(pictureFilename : String) : void
- +getDescription() : String
- +setDescription(description : String) : void
- +getCopies() : Integer
- +setCopies(copies : Integer) : void
- +getId() : String
- +setId(id : String) : void

AudioBook
- -format : Format
- +getFormat() : Format
- +setFormat(format : Format) : void

Format
- -description : String
- -additionalCost : float
- +getDescription() : String
- +setDescription(description : String) : void
- +getAdditionalCost() : float
- +setAdditionalCost(additionalCost : float) : void

CheckedOutMedia
- -media : Media
- -member : Member
- -due : Date
- +getMedia() : Media
- +setMedia(media : Media) : void
- +getMember() : Member
- +setMember(member : Member) : void
- +getDate() : Date
- +setDate(dueDate : Date) : void

WaitListMedia
- -media : Media
- -member : Member
- -status : String
- -expectedAvailability : Date
- +getMedia() : Media
- +setMedia(media : Media) : void
- +getMember() : Member
- +setMember(member : Member) : void
- +getStatus() : String
- +setStatus(status : String) : void
- +getExpectedAvailability() : Date
- +setExpectedAvailability(expectedAvailability : Date) : void

MediaDistributionSystem
- -members : ArrayList<Member>
- -admins : ArrayList<Admin>
- -name : String
- -checkedOut : ArrayList<CheckedOutMedia>
- -waitList : ArrayList<WaitListMedia>
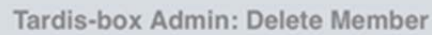- -media : ArrayList<Media>
- +getMembers() : ArrayList<Member>
- +setMembers(members : ArrayList<Member>) : void
- +getAdmins() : ArrayList<Admin>
- +setAdmins(admins : ArrayList<Admin>) : void
- +getName() : String
- +setName(name : String) : void
- +getCheckedOut() : ArrayList<CheckedOutMedia>
- +setCheckedOut(checkedOutMedia : ArrayList<CheckedOutMedia>) : void
- +getWaitList() : ArrayList<WaitListMedia>
- +setWaitList(waitList : ArrayList<WaitListMedia>) : void
- +getMedia() : ArrayList<Media>
- +setMedia(media : ArrayList<Media>) : void

Member

Admin

GenericUser
- -emailAddress : EmailAddress
- -firstName : String
- -lastName : String
- -passwordHash : String
- +getEmailAddress() : EmailAddress
- +setEmailAddress(emailAddress : EmailAddress) : void
- +getFirstName() : String
- +setFirstName(firstName : String) : void
- +getLastName() : String
- +setLastName(lastName : String) : void
- +getPasswordHash() : String
- +setPasswordHash(passwordHash : String) : void

EmailAddress
- -domainName : String
- -username : String
- +getUsername()
- +setUsername(username) : void
- +getDomainName() : String
- +setDomainName(domainName : String) : void
- +getUsername() : String
- +setUsername(username : String) : void
- +validate() : bool