**Password in table/database**

The passwords will be hashed before saving to the database. This means the password will be converted in to a indecipherable string of characters that are designed to be extremely difficult and close to impossible to decrypt and convert back in to a readable state. Hashing is a form of encryption designed not to be decrypted instead an entered password is hashed again with the same method and the resulting 2nd hash which is then compared to the original that is stored in the database. If they match then the access is granted, but the user created password in String format is never stored.

The way I plan to do this is using the Default PHP 5-5 “native password hashing API” which uses an implementation of BCrypt hashing, based on the Blowfish Cipher. For security the BCrypt Iterator takes a lot more time to try decrypt and harder to be deciphered by brute force attacks(see the sample code) Unlike the sample code I will not use the username as the salt or link the username to the hash because if the username is changed then the password would be required to be changed and hashed again with the new username as salt, or the username could not be changed at all.

Hashing method could not be used when storing credit card numbers for registered users, the ccv and expiry date would be requested to be input by the user at every transaction, they will never be stored. Although the encrypted credit card number that will be stored would be required to be decrypted to process the transaction without the user having to put it in every time.  
Although this may be unnecessary as the PayPal Payment Process using a PayPal account Login may have the users card linked. Therefore, there is little practical need for the user’s card to be stored. If storing credit card numbers is required, then it is best to use a trusted 3rd party company that is PCI certified or an official local Banking System that you can connect to, to store credit card data and process payments like the PayPal, Stripe or simple pay, as that is best and most secure Practice.

**User Characteristics and Roles**

1. **Registered User**The registered user will be able to access the application. When they register and save the details and agree to the terms and conditions they will have extra perks. Like choosing their favourite items and view their previous orders to make the order process faster. They will also gain extra discount codes or promotional content will be displayed to them upon log in.
2. **Unregistered User/Guest**The Guest User will be able to access and use the standard features like selecting items and purchasing them like registered users but will not be able to apply discount codes or see promotional content. They will be unable to save their favourite items.
3. **Administrator User**The Administrator user will have full access to the application as well as the administrator console view. Here they will manage Advertisements and Promotional Content Displaying. As well as manage Items for sale and Item Prices.  
   This will all be done through the administrator console view. In further development they may be able to generate reports based on anonymous data collected during the application use e.g. number if users, the type of users whether guest or registered, the Popular Items etc.

**System Architecture**

**Operating environment**

The Application is Designed to work on all web capable mobile devices. Specifically, for Smart Phones. The recommended browser is chrome although it will work on any web browser.  
The data will be stored on the server and the web application hosted online so a stable 4G or WIFI connection is required to access and use the web application.

**Development Environment**

The development will be done using a chrome browser, sublime Text editor and XAMPP localhost Server using myphpadmin for database creation and management. It will be developed in HTML 5, CSS 3, Bootstrap CSS, some JS elements (angular JS and React JS) and PHP 5.5.  
I will also be using the PayPal API payment gateway for Security and sandboxed payments in the Prototype to create a complete a complete user process or Journey.

**Assumptions and dependencies**

Without a WIFI or 3G/4G data connection users will not be able to connect to the internet to access the web application therefore an internet connection is required.

If a user doesn’t have either a credit/debit card or a PayPal account, they will be unable to purchase Items. An option for cash payment may be added in further developments, but for now Digital payments are required.

Users must consent to payment and accept terms and conditions of PayPal.

It is assumed that users are on premises of the cinema in in the screen and seat location they have input in to the application. In future development a JS seating map may be implemented.