CS 3610: Software Engineering

Spring 2014

Software Requirements Specification Document

Project Title

Theater Reservation System

Michael Hug hmichae4@students.kennesaw.edu
James Kimani jkimani2@students.kennesaw.edu
Justin Krynicki jkrynick@students.kennesaw.edu

Due Date: Wednesday 3/28/2014

Revision History

Date	Version	Description	Author
12-FEB-14	0.9	Initial Submission	
28-MAR-14	1.0	Final Draft	

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The Software Requirements Specification (SRS) provides and overview of the Theater Reservation System requirements. The SRS is the foundational document for how the Theater Reservation System will be constructed and utilized. Additionally, the SRS defines the relationships amongst various components and provides details of their interactions.

1.2 Scope

Currently, theater reservation systems horribly lag behind today's modern media advancements. Films are shown in the theaters using the latest technologies to grab audience attention; yet the very same reservation systems used to fill out the audience are outdated, unfriendly, and limited in their use. Theater reservation systems need to make available the entertainment, simplicity, and social integration that draw movie goers to films in the first place. The reservation system can be the movie-goer's first interaction with a film, and it is only beneficial to make the enjoyment and advancements of that experience as close to those of the film, itself. This is precisely where our Theater Reservation System can succeed and grab a large market share.

Potential theater customers will be able to access the theater system via Web page, kiosk, or app. The system will be able to provide not only reservations to the theater, but also reservations to specified seats if they are available. In order to assist and encourage undecided users in the movie selection process, the system will also provide suggestions, movie trailers, and user-generated content including, but not limited to, peer reviews, social media trends, and location sharing.

1.3 Definitions, Acronyms and Abbreviations

- SRS – Software Requirements Specification
- NFC - Near field communication
- ECMA - International association associated with the standardization of communication systems.
- Biometric – Identification of humans based on their biological uniqueness
- QR code – two dimension bar codes that can contains information or hyperlinks
- ITSC - Information Technology Support Center
- App – programs designed for mobile electronic devices, user authentication is occurs before installation and is persistent
- Kiosk – Free standing terminal that a person can use to access constrained software systems

1.4 References

- ECMA NFC specification - http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-352.pdf
- ITSC QR code specification - www.itsc.org.sg/pdf/synthesis08/Three_QR_Code.pdf

1.5 Overview

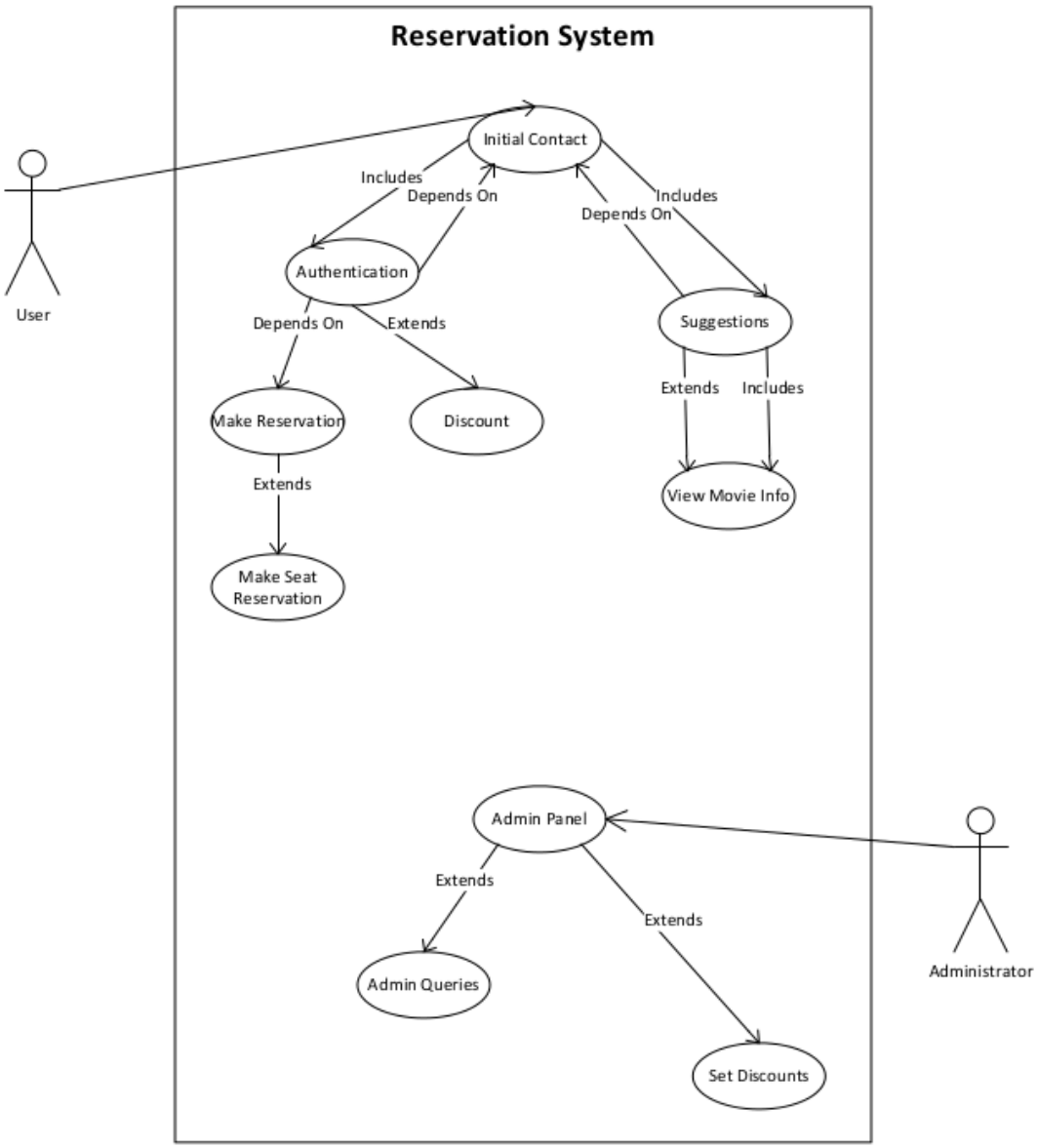
The remaining section of this document will contain the following :

- Diagram of Use-Cases (with description tables)
- Analysis classes (with description tables)
- UML Sequence Diagrams
- UML Behavior Diagrams

2. Overall Project Description

The objective of this project is the development of a web-based theater ticket reservation system for the ABC Entertainment Company. The product will allow customers to create reservations online for specified movie showings at all ABC Entertainment Company Theaters. The product will be accessible through internet-connected computers, tablets, and mobile devices. Customers will have the option to purchase their tickets at the time of reservation. If no purchase is made initially, customers may choose to pick-up tickets at the specified theater's box office. With this option, customers will have until thirty minutes prior to schedule show time to claim their reservations in-person at the theater and complete their purchase. If the reservation is not claimed thirty minutes prior to schedule show time, the reservation will be canceled and the allotted seats will be released for purchase by the general public. Customers may also choose to complete their purchase for previously reserved seats online and print their tickets at home up to one hour prior to schedule show time. If customers choose the print option and do not complete purchase by the specified time, the reservation will be canceled and the seats will be released to the general public.

2.1 Use-Case Model Survey


Reservation System
User
Initial Contact
Includes
Depends On
Includes
Depends On
Authentication
Suggestions
Depends On
Extends
Extends
Includes
Make Reservation
Discount
View Movie Info
Extends
Make Seat Reservation
Admin Panel
Extends
Extends
Admin Queries
Set Discounts
Administrator

Use case name: Initial contact	**ID:** 001	**Priority:** High
Primary actor: Unknown user	**Source:** User	**Use case type:** Technical
Interested Stakeholders: Movie Theater		
Brief description: An unknown user opens the app / requests the web site / engages a kiosk. The user is then given the option to authenticate or continue anonymously		
Precondition: App / web site / kiosk is in a ready state		
Trigger: An unknown user opens the app / requests the web site / engages a kiosk		
Relationships: **Include:** Suggestions, View detailed movie information **Extends:** None **Depends On:** None		
Typical flow of events: 1. An unknown user opens the app / requests the web site / engages a kiosk 1. The user chooses to authenticate 1. Current flow is terminated and ID 003 is initiated 2. The user chooses to not authenticate 1. Current flow is terminated and ID 002 is initiated		
Assumptions: App / web site / kiosk can connect to the theater server through a secure connection		
Implementation Constraints and Specifications: Apps may be left in an authenticated state. Apps will always have initial contact as the first step, but if the user never logs out, the app stays authenticated.		

Use case name: Suggestions	**ID:** 002	**Priority:** High
Primary actor: Anonymous user	**Source:** User	**Use case type:** Technical
Interested Stakeholders: Movie Theater		
Brief description: Suggestions are generated by the system and displayed to the user		
Precondition: None		
Trigger: User opts to view suggestions		
Relationships: **Include:** View detailed movie information **Extends:** None **Depends On:** Initial contact		
Typical flow of events: 1. Suggestions are made based on prior anonymous data collected at that location or user data 1. The location will be relative, multiple kiosks at the same postage address will be considered the same location 2. Mobile location data will be considered the same location if it is within 10 miles 3. Locations will overlap and share relative data 2. User is given the option to view detailed movie information. 3. User is given the option to view more suggestions		
Assumptions: The GUI is designed in a manner that an option to authenticate or make a reservation is always available		
Implementation Constraints and Specifications: The options to authenticate or make a reservation are mutually exclusive		

Use case name: View detailed movie information	**ID:** 003	**Priority:** Low
Primary actor: User	**Source:** User	**Use case type:** Technical

Interested Stakeholders: Movie Theater

Brief description: A user opts to view detailed movie information

Precondition: Suggestions have been shown

Trigger: User clicked on suggestion

Relationships:
- **Include:** None
- **Extends:** Suggestions
- **Depends On:** None

Typical flow of events:

1. The selection of the movie is kept in a database and associated with either the user or location
2. A screen is displayed where
    1. User can authenticate to make a reservation or add a comment
        1. If the user is already authenticated, this option is now to make reservation
    2. User can view suggestions
    3. User can watch movie trailer
    4. User can view movie synopsis
    5. User can view user generated comments
        1. If user is authenticated, user can add comment

Assumptions: The movie details are interesting enough for the user to want to then make a movie reservation

Implementation Constraints and Specifications:
The user may enter an infinite loop of viewing detailed information and suggestions

Use case name: Authentication	**ID:** 004	**Priority:** High
Primary actor: User	**Source:** User	**Use case type:** Technical
Interested Stakeholders: Movie Theater		
Brief description: A user will attempt to authenticate with the system		
Precondition: User have selected the option to authenticate		
Trigger: User has selected the option to authenticate		
Relationships: **Include:** Initial contact **Extends:** Discount **Depends On:** None		
Typical flow of events: 1. Authentication options will be given based on the device used to authenticate 2. User attempts to authenticate 1. If authentication is unsuccessful 1. the user is given the option to restart authentication, this will force a new check of environmental variables 2. The user is given the option to cancel authentication 2. If authentication is successful 1. The system checks for a discount 2. The user is given the option to view suggestions		
Assumptions: Every app / computer / kiosk will allow at least user name / password authentication, devices have environmental variables that will be checked at runtime. These environmental variables are handled at the operating system level thus they are abstracted from this software		
Implementation Constraints and Specifications: Mobile devices, computers and kiosk technology are advancing constantly. Every device will have the software available to authenticate with all three methods.		

Use case name: Discount	**ID:** 005	**Priority:** Low
Primary actor: User	**Source:** User	**Use case type:** Technical

Interested Stakeholders: Movie Theater

Brief description: A user is offered a discount

Precondition: User is authenticated

Trigger: User authentication

Relationships:

Include: None
Extends: Authentication
Depends On: None

Typical flow of events:

1. Authentication options will be given based on the device used to authenticate
2. User attempts to authenticate
    1. If authentication is unsuccessful
        1. the user is given the option to restart authentication, this will force a new check of environmental variables
        2. The user is given the option to cancel authentication
    2. If authentication is successful
        1. The system checks for a discount
        2. The user is given the option to view suggestions
        3. The user is given the option to make reservation

Assumptions: Discounts will be offered globally, based on location or based on user

Implementation Constraints and Specifications:

The user is given a discount at one time and must enter the discount at payment time. Payment is outside the scope of this software.

Use case name: Make reservation	**ID:** 006	**Priority:** High
Primary actor: User	**Source:** User	**Use case type:** Technical
Interested Stakeholders: Movie Theater		
Brief description: A user wants to make a reservation		
Precondition: User is authenticated		
Trigger: User selects the option to make reservation		
Relationships: **Include:** None **Extends:** None **Depends On:** Authentication		
Typical flow of events: 1. User opts to make a reservation 2. List of movies available are displayed 3. User clicked on movie to reserve 4. User is given option to reserve seat 5. User is given option to view suggestions		
Assumptions: System is secure and working		
Implementation Constraints and Specifications: The list of movies available in theaters is small enough to be displayed.		

Use case name: Make seat reservation	**ID:** 007	**Priority:** Low
Primary actor: User	**Source:** User	**Use case type:** Technical
Interested Stakeholders: Movie Theater		
Brief description: A user wants to make a seat reservation		
Precondition: User has a reservation		
Trigger: User has made reservation and opts to reserve a seat		
Relationships: **Include:** None **Extends:** Make reservation **Depends On:** None		
Typical flow of events: 1. User opts to make a seat reservation 2. List of seats available are displayed 3. User clicked on seat to reserve 4. User is given option to view suggestions		
Assumptions: There is a seat available		
Implementation Constraints and Specifications: The list of seats available in theaters is small enough to be displayed.		

Use case name: Admin panel	**ID:** 008	**Priority:** Low
Primary actor: Administrator	**Source:** Administrator	**Use case type:** Technical

Interested Stakeholders: Movie Theater

Brief description: An administrator wants to log on to the administrator panel

Precondition: None

Trigger: Administrator connects via the admin terminal

Relationships:

Include: Set discounts
Extends: None
Depends On: None

Typical flow of events:

1. Administrator connect via a secure ssh connection to the administrator panel
2. Administrator panel is command line only
3. The administrator panel authenticates the administrator using only a SHA-256 fingerprint
    1. Passwords or any other type of authentication will not be available.
        1. Remote connections will only connect with SHA fingerprint
        2. Physical access to the server will be securely restricted to authorized personnel
4. Administrator can query the list of all user generated and anonymous generated data
5. Administrator sets up discount trigger

Assumptions: Physical access to the server is only used to add remote admin access to the server

Implementation Constraints and Specifications:

Physical access to the server is securely restricted

Use case name: Set discounts		**ID:** 009	**Priority:** Low
Primary actor: Administrator	**Source:** Administrator		**Use case type:** Technical
Interested Stakeholders: Users			
Brief description: An administrator is able to set discount triggers			
Precondition: None			
Trigger: Administrator opens discount interface			
Relationships: **Include:** None **Extends:** Admin panel **Depends On:** None			
Typical flow of events: 1. Administrator will set discounts based on location or user 2. Location discounts will be be available to every user in a location 3. User discounts will only be extended to groups of users			
Assumptions: Marketing department has given administrator an order for the system to offer discounts			
Implementation Constraints and Specifications: None			

Use case name: Admin queries	**ID:** 010	**Priority:** Low
Primary actor: Administrator	**Source:** Administrator	**Use case type:** Technical
Interested Stakeholders: Marketing department		
Brief description: An administrator can run SQL queries for the marketing department		
Precondition: Administrator is logged on		
Trigger: Administrator enters a SQL query		
Relationships: **Include:** None **Extends:** Admin panel **Depends On:** None		
Typical flow of events: 1. Administrator enters the SQL command line interface. 2. Administrator runs query 3. Data is returned to the marketing department however it is desired 1. Administrator has the option to verbally return data, generate reports, output CSV files		
Assumptions: Marketing department wants data from the server		
Implementation Constraints and Specifications: The administrator will return raw that to the marketing department. The marketing department is responsible for all data analysis, interpretation and presentation.		

2.2 Assumptions and Dependencies

- The product must be delivered by the deadline.
- The budget cannot be exceeded.
- Customer information must be secured and held to the strictest privacy constraints.
- The product must be easily accessible for all users with an internet-connected device.
- The product must be user-friendly and reliable.
- The product must have forward compatibility.
- App / web site / kiosk can connect to the theater server through secure connections.
- Apps may be left in an authenticated state. Apps will always have initial contact as the first step, but if the user never logs out, the app stays authenticated.
- The GUI is designed in a manner that an option to authenticate or make a reservation is always available.
- The options to authenticate or make a reservation are mutually exclusive.
- The movie details are interesting enough for the user to want to then make a movie reservation.
- The user may enter an infinite loop of viewing detailed information and suggestions.
- Every app / computer / kiosk will allow at least user name / password authentication, devices have environmental variables that will be checked at run-time. These environmental variables are handled at the operating system level thus they are abstracted from this software.
- Mobile devices, computers and kiosk technology are advancing constantly. Every device will have the software available to authenticate with all three methods.
- Discounts will be offered globally, based on location or based on user.
- The user is given a discount at one time and must enter the discount at payment time. Payment is outside the scope of this software.
- System is secure and working.
- The list of movies available in theaters is small enough to be displayed.

- Physical access to the server is only used to add remote admin access to the server.
- Physical access to the server is securely restricted.
- Marketing department has given administrator an order for the system to offer discounts.
- The marketing department asks for the data in a reasonable manner.
- The administrator will return raw that to the marketing department. The marketing department is responsible for all data analysis, interpretation and presentation.

3. Specific Requirements

External Interface Requirements

User Interfaces

The project is designed to be an application that can be accessed from several different types of devices. A user will access the application via a kiosk, app, or web browser. Each device will be only a front end to the same behind the scenes work. Each user interface will access the same content.

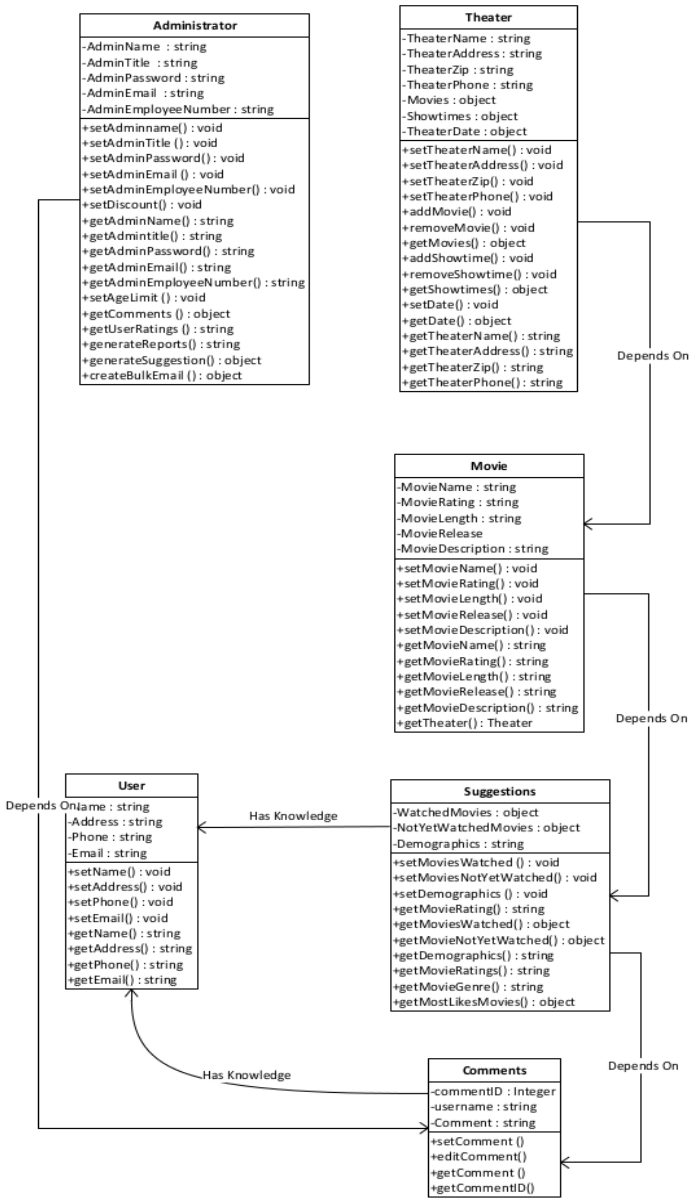
Hardware Interfaces

Servers will be necessary to host the system. Most servers will need to be added as the theater system grown and expands with user content and user data.

Software Interfaces

The software will be designed to be used by an app, a website, or through a kiosk. Each interface will appear the same to the user and no different to the system.

3.1 Classes/Objects



Class Name: User	
Description: Representation of a user which will contain individual user information required for authentication.	
Attributes (fields)	**Attribute Description**
Private string Name	Holds name of user
Private string Address	Holds address of user
Private string Phone	Holds phone 10-digit phone number of user (allows the entry of hyphens)
Private string Email	Holds email address of user
Methods (operations)	**Method Description**
Public void setName	Sets user name
Public void setAddress	Sets user address
Public void setPhone	Sets user phone
Public void setEmail	Sets user email
Public string getName	Returns user name
Public string getAddress	Returns user address
Public string getPhone	Returns user phone
Public string getEmail	Returns user email

Class Name: Theater	
Description: Contains information on a theater	
Attributes (fields)	**Attribute Description**
Private string TheaterName	Holds the theater name
Private string TheaterAddress	Holds the theater address
Private string TheaterZip	Holds the zip code of a theater
Private string TheaterPhone	Holds the theater phone number
Private list Movies	Holds the set of all movies showing at the theater on a present or future date
Private list Showtimes	Holds the set of show times of a selected movie on a given date

Private Date TheaterDate	Holds the date to search within for movies playing in the theater
Methods (operations)	**Method Description**
Public void setTheaterName	Sets theater name
Public void setTheaterAddress	Sets theater address
Public void setTheaterZip	Sets theater zip
Public void setTheaterPhone	Sets theater phone number
Public void addMovie	Adds movie to the theater
Public void removeMovie	Removes movie from the theater
Public list getMovies	Returns the list of all movies at the theater
Public void addShowtime	Adds show time to a movie at the theater
Public void removeShowtime	Removes show time to a movie at the theater
Public list getShowtimes	Returns the list of show times for a movie at the theater
Public void setDate	Sets date for user to search within
Public Date getDate	Returns the date to search within
Public string getTheaterName	Returns theater name
Public string getTheaterAddress	Returns theater address
Public string getTheaterZip	Returns theater zip
Public string getTheaterPhone	Returns theater phone number

Class Name: Administrator	
Description: Defines administrator responsibilities and capabilities	
Attributes (fields)	**Attribute Description**
Private String AdminName	Holds the name of the Admin
Private String AdminTitle	Holds the Admin title or position in the company
Private String AdminPassword	Holds the admin password
Private String AdminEmail	Holds the admin email
Private String AdminEmployeeNumber	Hold the admin employee ID number
Methods (operations)	**Method Description**

Public void setAdminname	Set the admin name
Public void setAdminTitle	Set the admin Title
Public void setAdminPassword	Set the admin password
Public void setAdminEmail	Set the admin email
Public void setAdminEmployeeNumber	Set the admin employee ID number
Public void setDiscount	Set the discount rate or value
Public string getAdminName	Return the admin Name
Public string getAdmintitle	Return admin Title
Public string getAdminPassword	Return the admin password
Public string getAdminEmail	Return the admin Email
Public string getAdminEmployeeNumber	Return the admin Employee ID number
Public void setAgeLimit	Sets the age limit for movie goers based on the MPAA rating
Public comment getComments	Returns the user comments of a movie
Public string getUserRatings	Returns the user ratings
Public string generateReports	Returns the generalized report of a movie
Public suggestion generateSuggestion	Returns suggestions of movies based on user's previously watched movies, demographics and unwatched movies
Public email createBulkEmail	Return a bulk email to users about new movies showing or suggestions

Class Name: Movie	
Description: Contains information on a movie	
Attributes (fields)	**Attribute Description**
Private string MovieName	Holds the name of the movie
Private string MovieRating	Holds the MPAA rating of the movie
Private string MovieLength	Holds the running time of the movie
Private string MovieRelease	Holds the release date of the movie
Private String MovieDescription	Hold a brief description of the movie
Methods (operations)	**Method Description**

Public void setMovieName	Sets the name of the movie
Public void setMovieRating	Sets the rating of the movie
Public void setMovieLength	Sets the running time of the movie
Public void setMovieRelease	Sets the release date of the movie
Public void setMovieDescription	Sets the description of the movie
Public String getMovieName	Returns the name of the movie
Public string getMovieRating	Returns the rating of the movie
Public string getMovieLength	Returns the running time of the movie
Public string getMovieRelease	Returns the release date of the movie
Public string getMovieDescription	Returns the description of the movie
Public Theater getTheater	Returns a designated theater

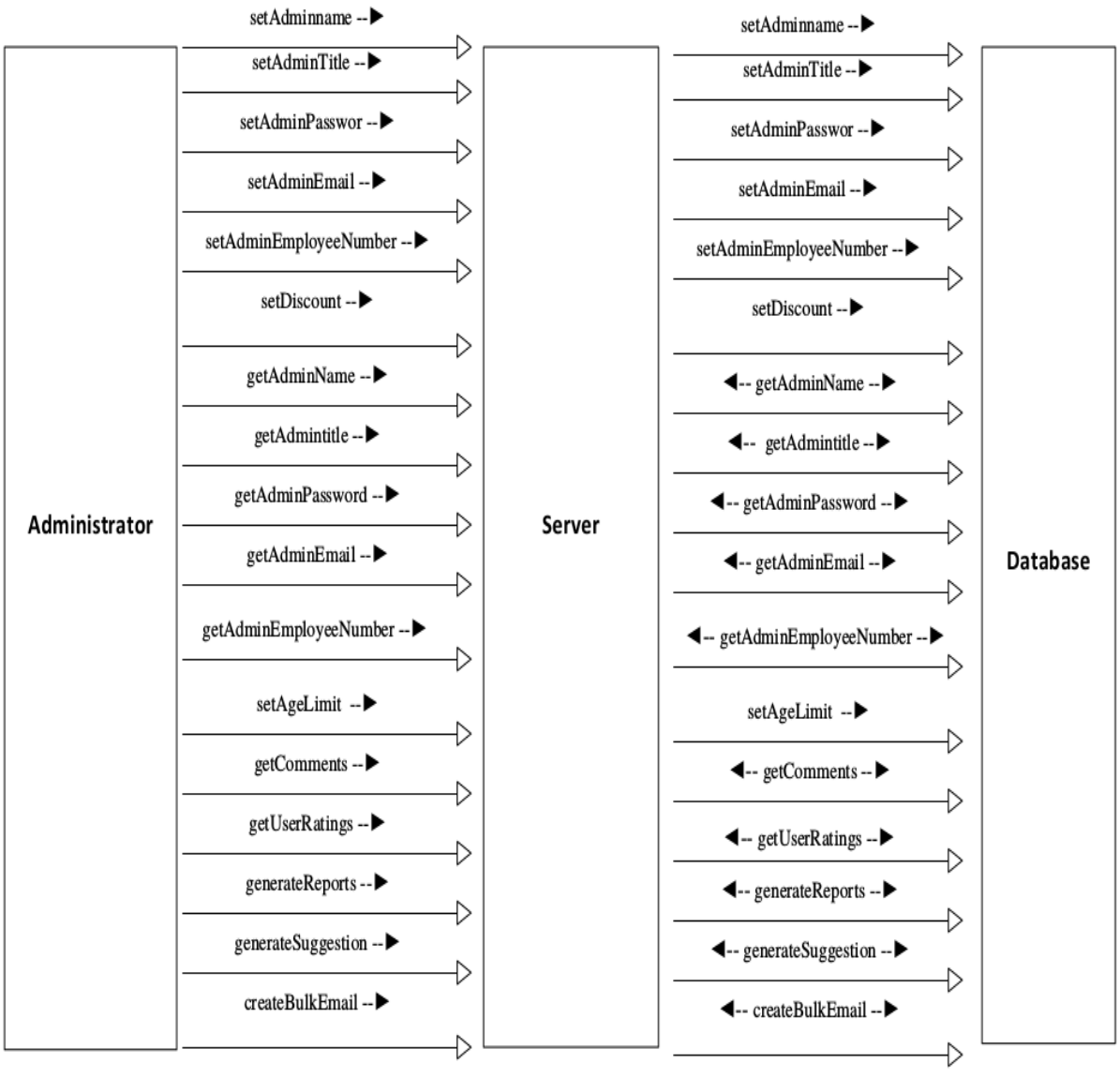
Class Name: Comments	
Description: Contains of a movie	
Attributes (fields)	**Attribute Description**
Private integer commentID	Holds the auto generated comment ID
Private string username	Holds the commenter's username or screenName
Private string Comment	Holds the comment from end-user
Methods (operations)	**Method Description**
Public void setComment	Sets the comment for a movie
Public void editComment	Sets the new comment for a movie and replaces the old one from same user
Public string getComment	Returns the comment for a movie
Public interger getCommentID	Return the auto-generated commentID

Class Name: Suggestions	
Description: Contains suggestions of a movie based on previously watched movies and demographics	
Attributes (fields)	**Attribute Description**
Private list WatchedMovies	Holds the names of the previously watched movies

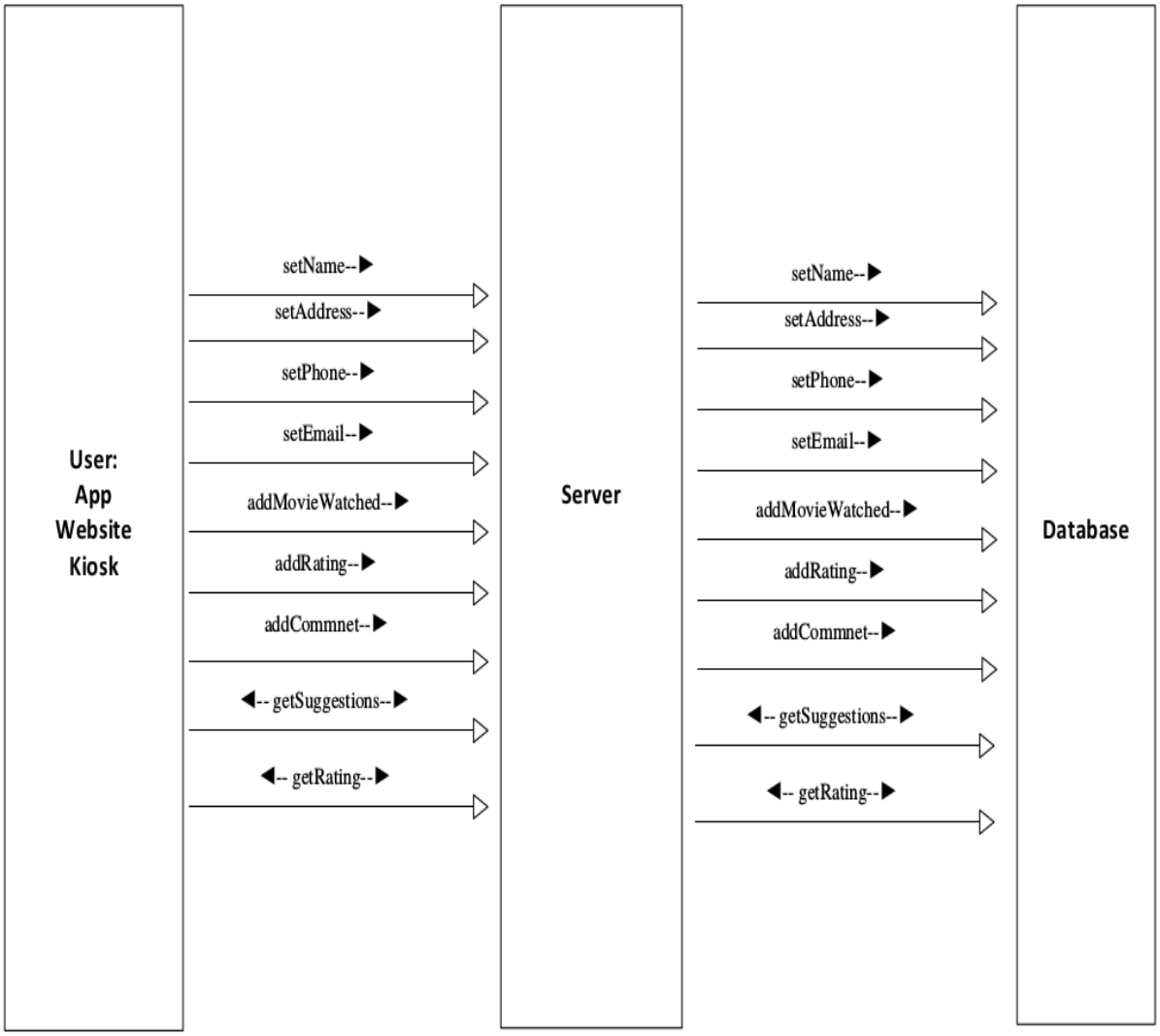
Private list NotYetWatchedMovies	Holds the name of movies of movies not yet watched
Private string Demographics	Holds the user's demographic information
Methods (operations)	**Method Description**
Public void setMoviesWatched	Sets the name of the movies watched
Public void setMoviesNotYetWatched	Sets the movies not yet watched
Public void setDemographics	Sets the user demographics
Public string getMovieRating	Returns the rating of the movie
Public list getMoviesWatched	Returns a List of already watched movies
Public list getMovieNotYetWatched	Returns a List of not yet watched movies
Public string getDemographics	Returns the user Demographics
Public string getMovieRatings	Returns the movie rating
Public string getMovieGenre	Returns the movie
Public list getMostLikesMovies	Return a list of user's most liked movies or movies with good ratings that the user has not watched yet

3.2 Object Collaboration Diagrams

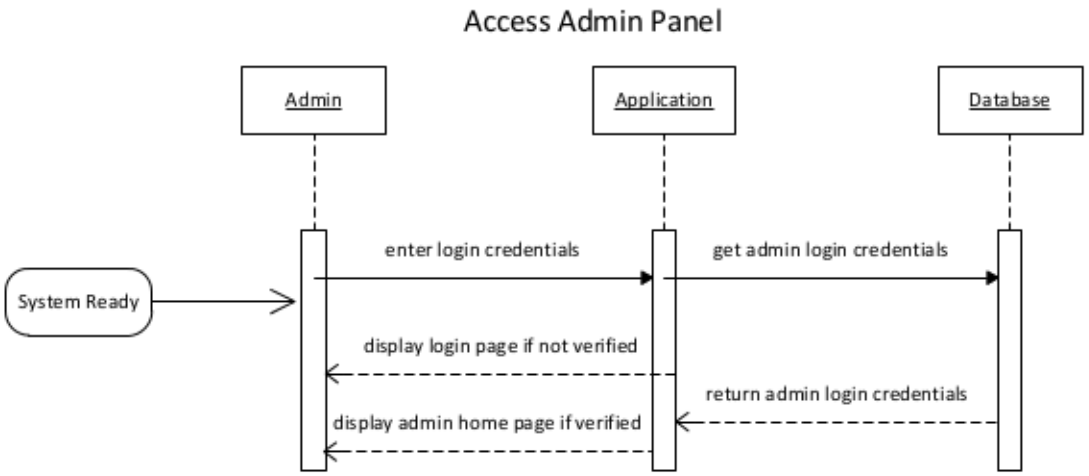
Administrator Collaboration

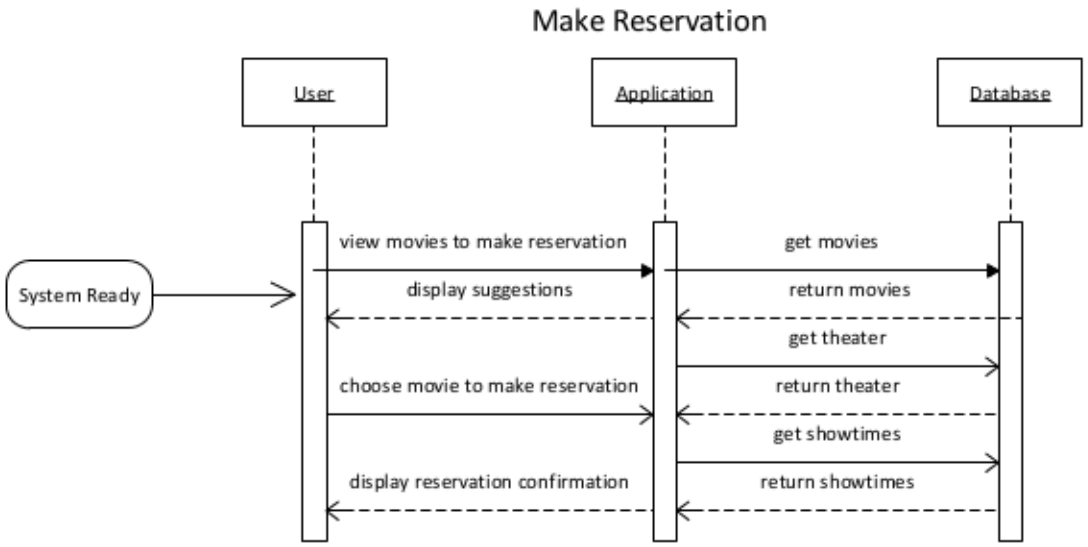

Administrator
Server
Database
setAdminname --▶
setAdminTitle --▶
setAdminPasswor --▶
setAdminEmail --▶
setAdminEmployeeNumber --▶
setDiscount --▶
getAdminName --▶
getAdmintitle --▶
getAdminPassword --▶
getAdminEmail --▶
getAdminEmployeeNumber --▶
setAgeLimit --▶
getComments --▶
getUserRatings --▶
generateReports --▶
generateSuggestion --▶
createBulkEmail --▶
setAdminname --▶
setAdminTitle --▶
setAdminPasswor --▶
setAdminEmail --▶
setAdminEmployeeNumber --▶
setDiscount --▶
◀-- getAdminName --▶
◀-- getAdmintitle --▶
◀-- getAdminPassword --▶
◀-- getAdminEmail --▶
◀-- getAdminEmployeeNumber --▶
setAgeLimit --▶
◀-- getComments --▶
◀-- getUserRatings --▶
◀-- generateReports --▶
◀-- generateSuggestion --▶
◀-- createBulkEmail --▶

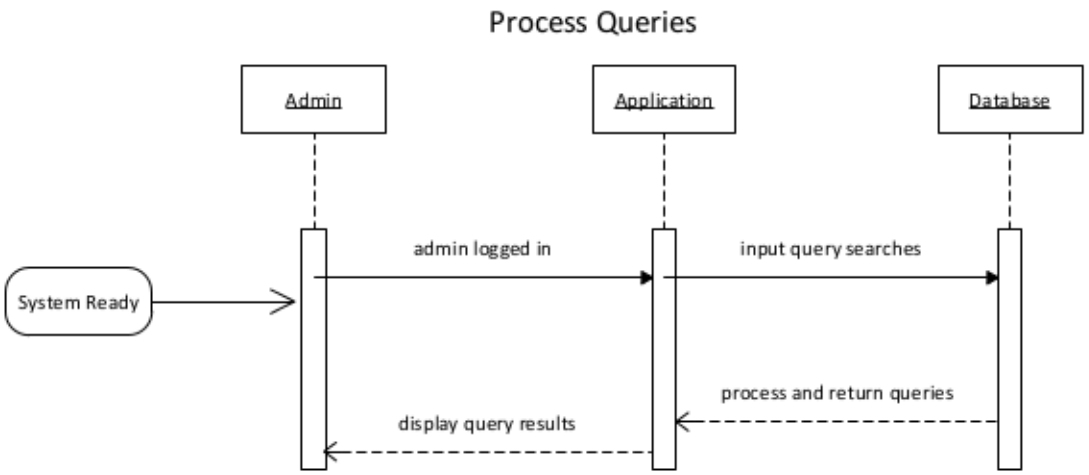
User Collaboration

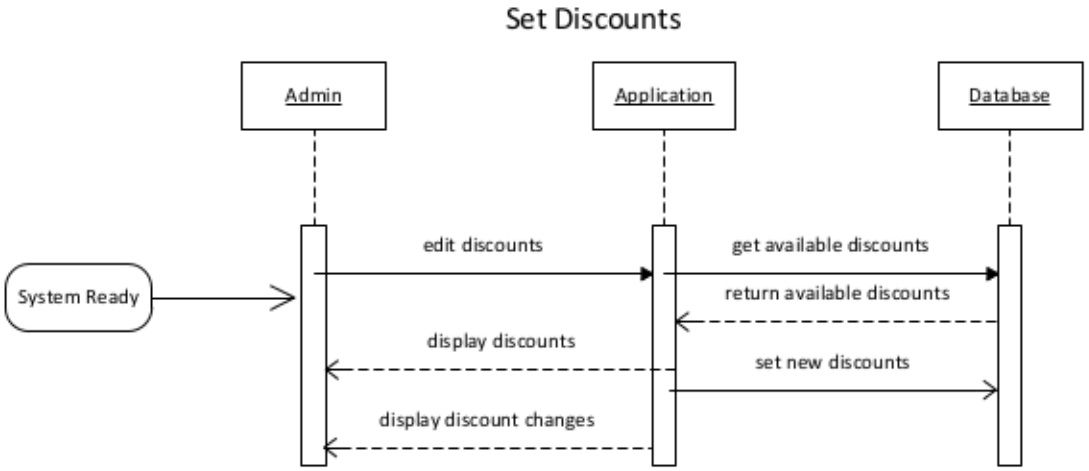


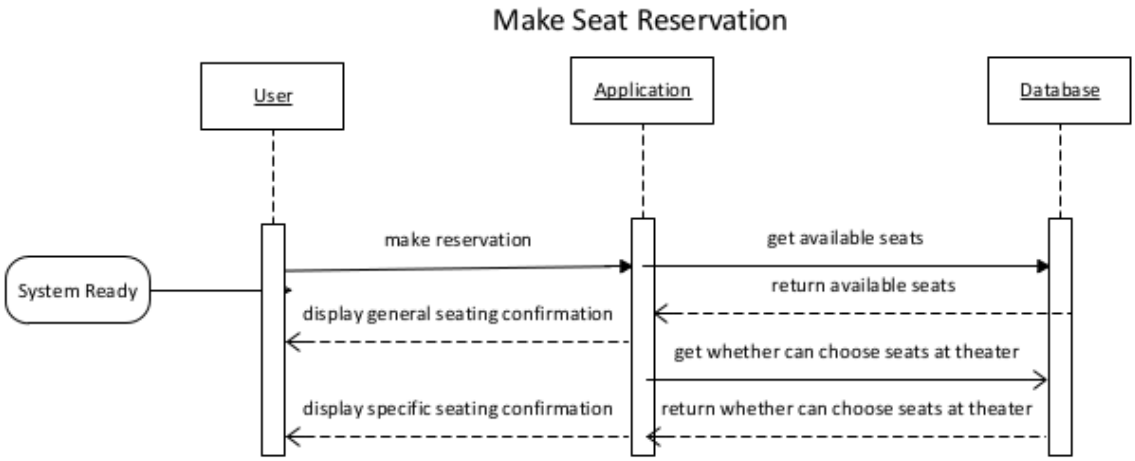
3.3 Sequence Diagrams

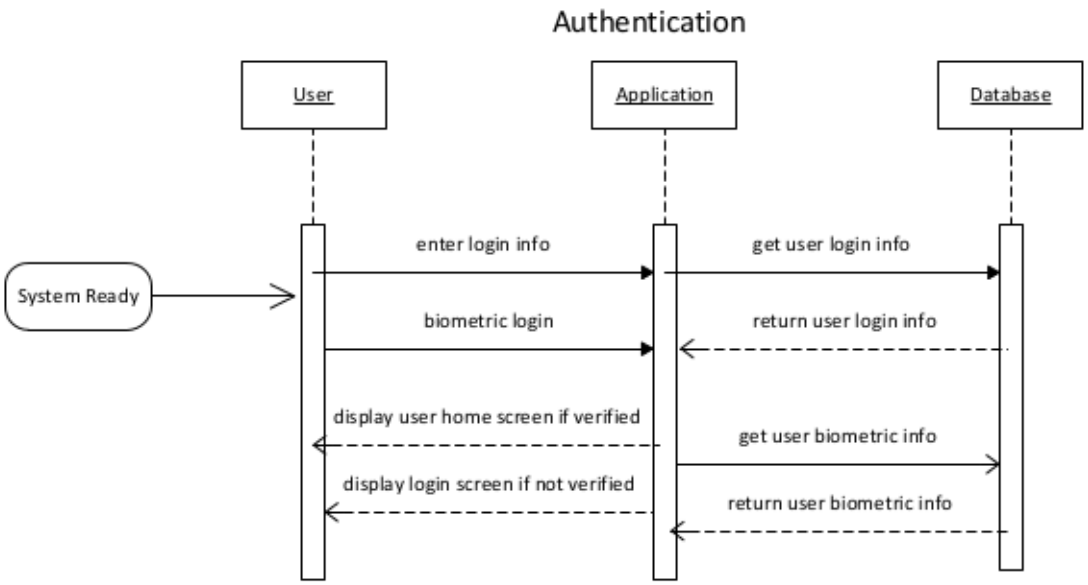


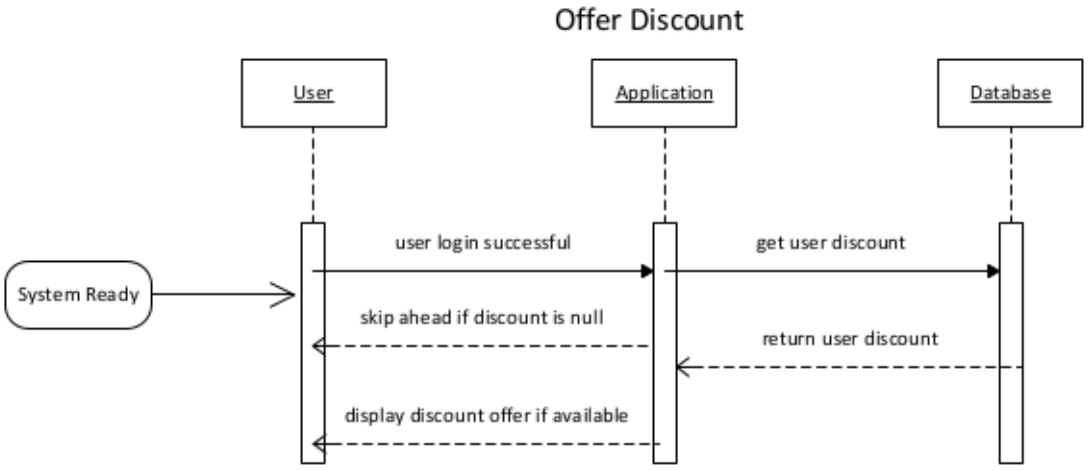

Make Reservation
User
Application
Database
System Ready
view movies to make reservation
display suggestions
choose movie to make reservation
display reservation confirmation
get movies
return movies
get theater
return theater
get showtimes
return showtimes

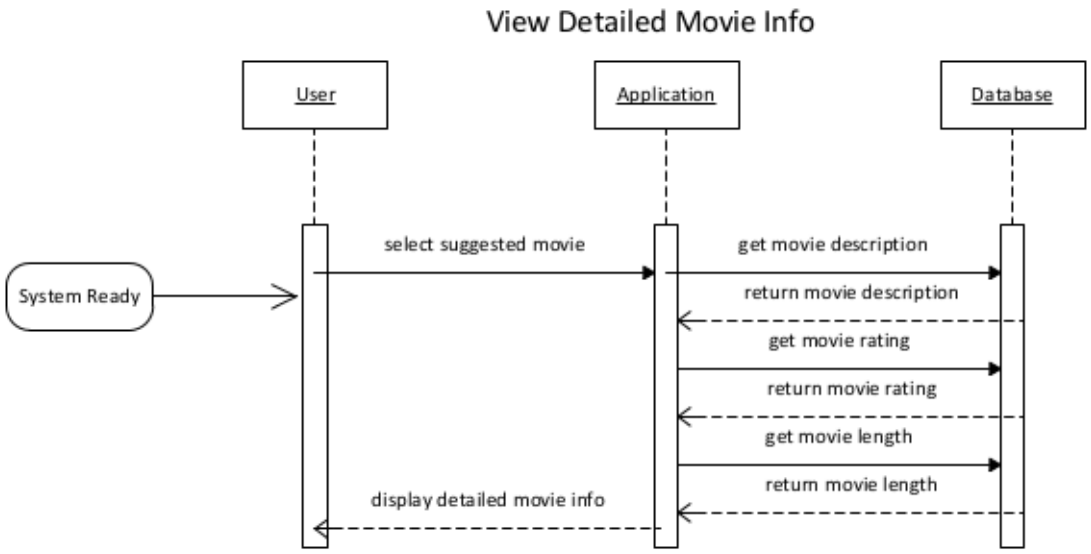

Process Queries
Admin
Application
Database
admin logged in
input query searches
System Ready
process and return queries
display query results

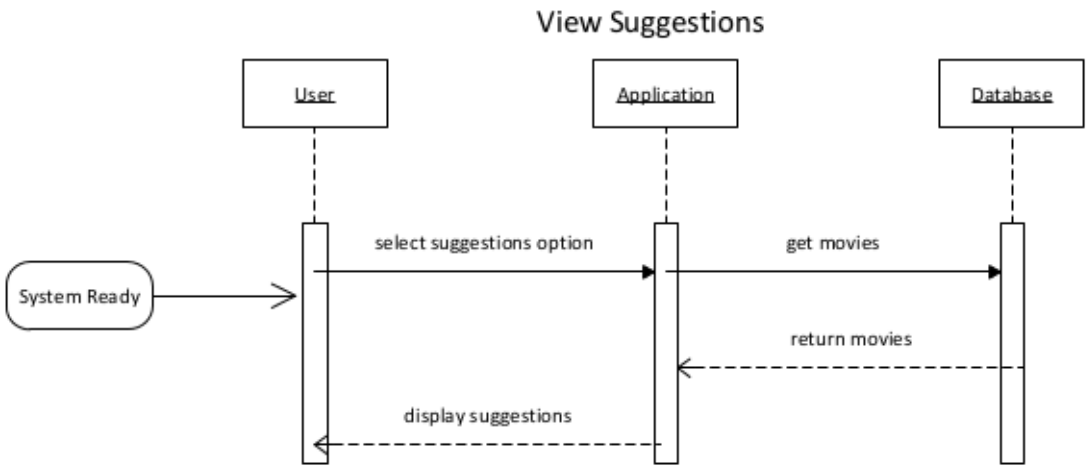

Set Discounts
Admin
Application
Database
System Ready
edit discounts
get available discounts
return available discounts
display discounts
set new discounts
display discount changes


Make Seat Reservation
User
Application
Database
System Ready
make reservation
get available seats
return available seats
display general seating confirmation
get whether can choose seats at theater
display specific seating confirmation
return whether can choose seats at theater

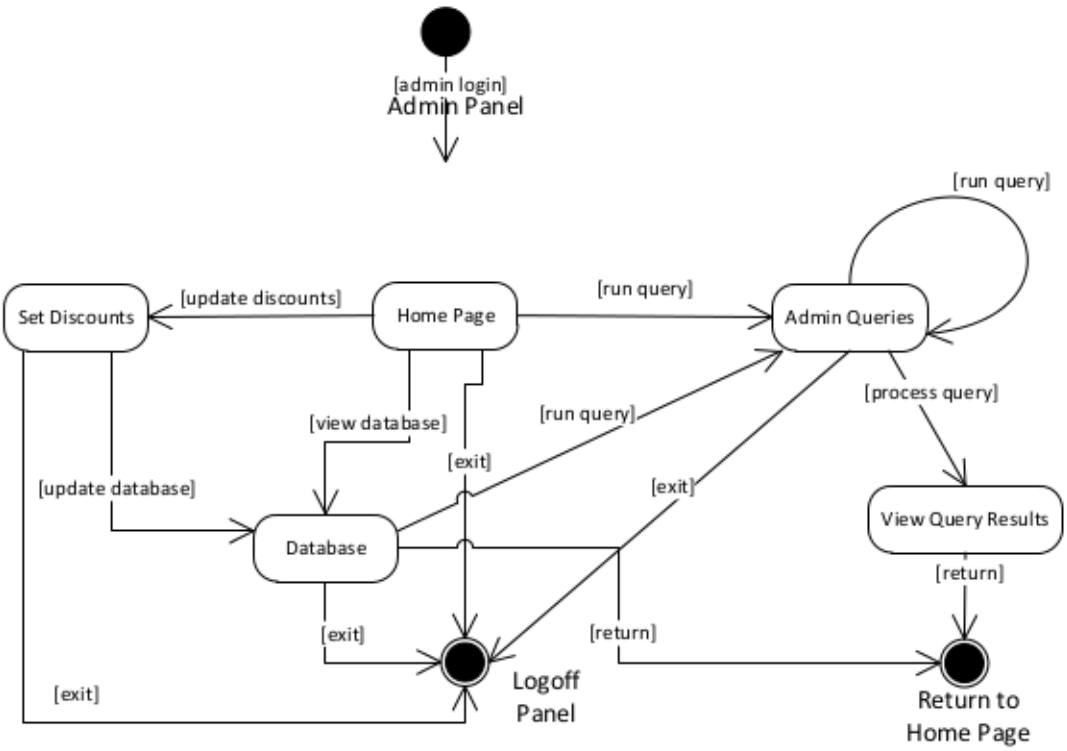

Authentication
User
Application
Database
System Ready
enter login info
get user login info
biometric login
return user login info
display user home screen if verified
get user biometric info
display login screen if not verified
return user biometric info

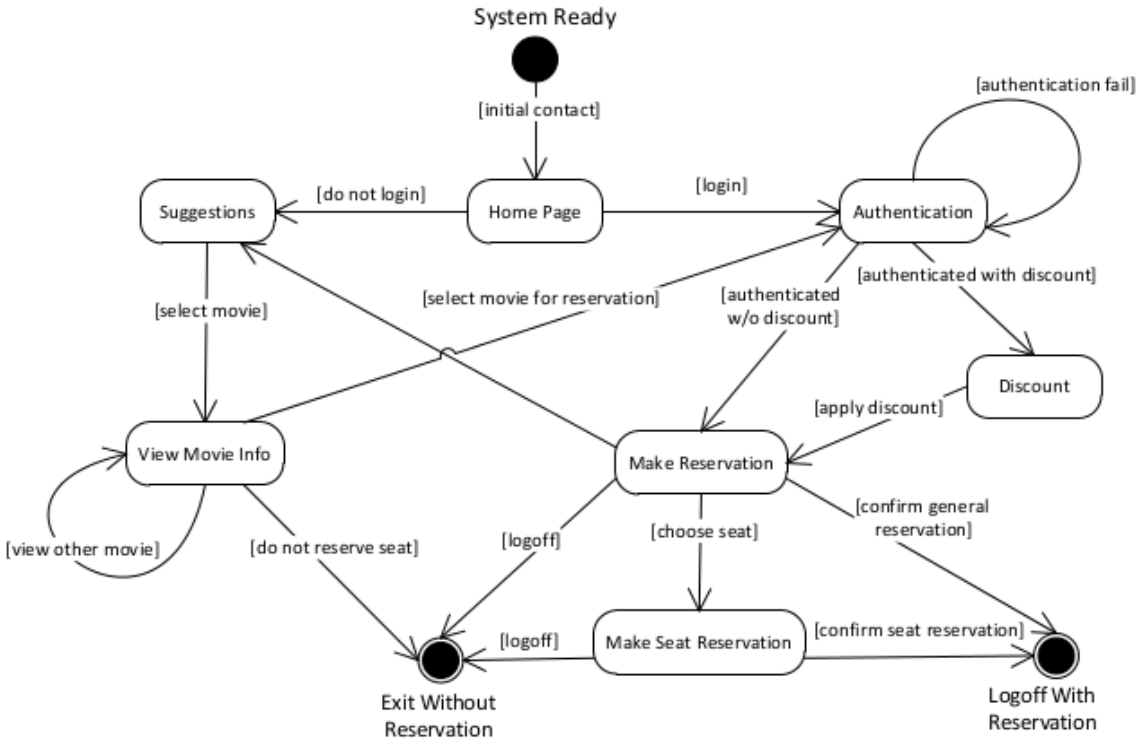

Offer Discount
User
Application
Database
System Ready
user login successful
get user discount
skip ahead if discount is null
return user discount
display discount offer if available


View Detailed Movie Info
User
Application
Database
System Ready
select suggested movie
get movie description
return movie description
get movie rating
return movie rating
get movie length
return movie length
display detailed movie info


View Suggestions
User
Application
Database
System Ready
select suggestions option
get movies
return movies
display suggestions

3.4 Object Behavior Diagrams





3.5 Performance Requirements

- The kiosk need to be NFC enabled and have biometric reader.
- The database must be secure.
- The database needs to be indexed and cache enabled for faster query rate times
- Scheduled server maintenance must be made in a manner that the server operations are not disrupted.
- The web server must offer SSL for authentication.
- The server must have ample storage space.
- The server must be capable of handing high traffic

3.6 Other Requirements

- The chairs in the theater must be clearly marked.
- The staff will be required to put a card on the seats that become reserved.

4. Supporting Information

1. http://en.wikipedia.org/wiki/Near_field_communication
2. http://newtech.about.com/od/mobile/a/Near-Field-Communications-NFC.htm
3. http://en.wikipedia.org/wiki/Biometrics
4. http://technet.microsoft.com/en-us/library/ms171908(v=sql.90).aspx
5. http://blogs.telerik.com/kendoui/posts/13-09-04/creating-a-mobile-site-for-your-sql-server-data
6. http://developer.android.com/index.html
7. https://developer.apple.com/technologies/ios/
8. http://msdn.microsoft.com/en-us/library/aa479073.aspx