

CS 3610: Software Engineering

Spring 2014

Software Design Document (SDD)

Group # 3

Theater Reservation System

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

Due Date: Wednesday 4/23/2014

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

Revision History

Date	Version	Description	Author
23-April-14	0.9	Initial Submission	Hug, Kimani, Krynicky

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.4	References	4
2.	Object Descriptions	5
2.1	Objects Class Diagram	5
2.2	Objects in Subsystem	6
2.3	Pseudocode	Error! Bookmark not defined.
3.	Sequence Diagrams	14
4.	Object Collaboration	21
5.	Object behavior	23
6.	Data Design and Database Design	24
7.	Other Design Requirements	26

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

Software Requirements Specification

1. Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software

Design Document(SDD) are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information.

1.1 Purpose

The purpose of the software Design Document (SDD) is to provide a description of the design of a system fully enough to allow for software development to proceed with and understanding of what is to be necessary to provide a description of the details for the software and system to be built.

1.2 Scope

This Software Design Document is for a base level system which will work as a proof of concept for the use of building a system provides a base level of functionality to show feasibility for large scale production use. This Software Design is focused on the base level system and critical parts of the system. For this particular Software Design Document (SDD), the focus is placed on generation of the documents and modification of the documents. The system will be used in conjunction with other pre-existing systems and will consist largely of a document interaction facade that abstracts document interactions and handling of the document objects.

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

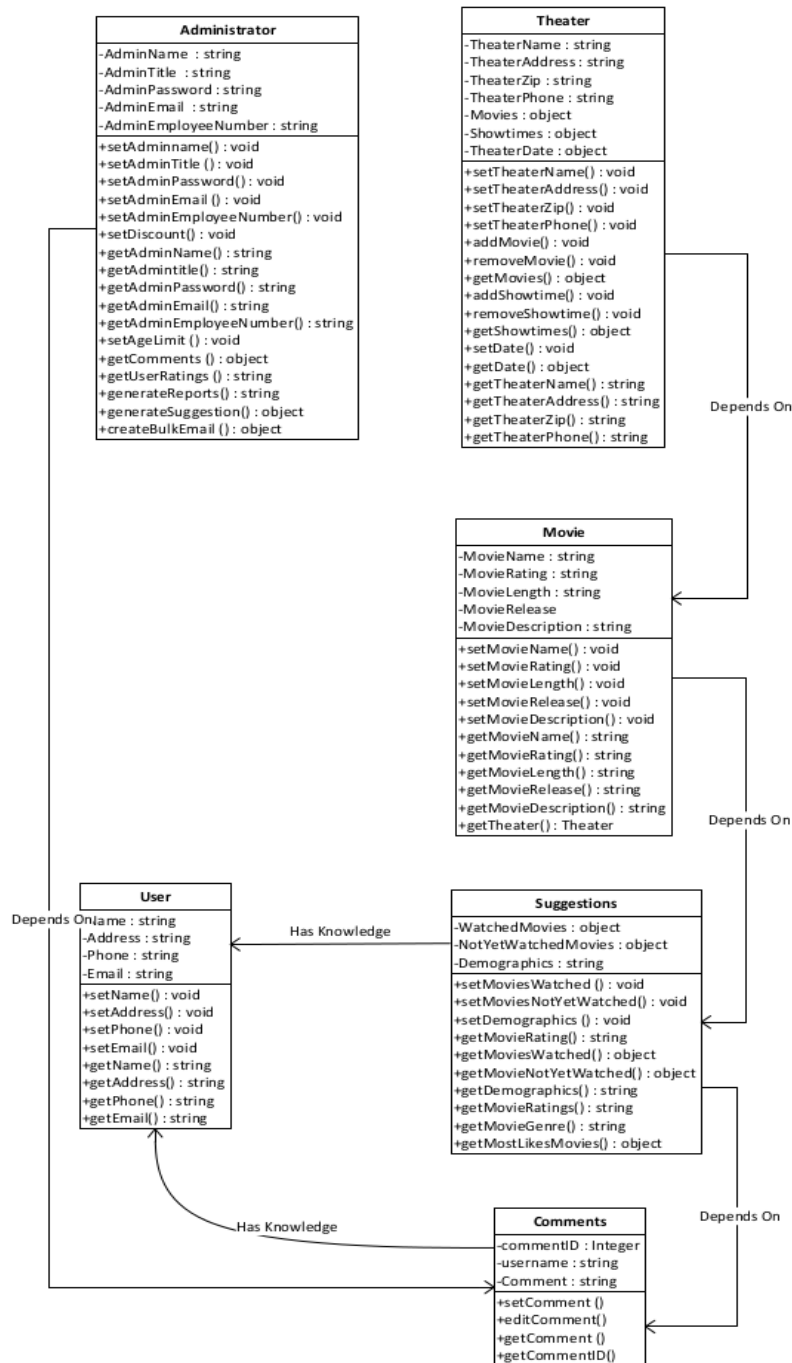
1.4 References

- 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

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

2. Object Descriptions

2.1 Objects Class Diagram



Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

2.2 Objects in Subsystem

Class Name: User				
Description: Representation of a user which will contain individual user information required for authentication.				
Attribute	Data Type	Access Level	Default	Description
Name	String Length 1 ... 25	Private	"Anonymous"	Holds name of user
Address	String Length 1 ... 25	Private	"123 ABC Place Kennesaw GA, 30144"	Holds address of user
Phone	String Length 10	Private	"000-000-0000"	Holds phone 10-digit phone number of user (allows the entry of hyphens)
Email	String Length 7...32	Private	"Anonymous@abc.com"	Holds email address of user
Methods	Return Type	Access Level	Restrictions	Description
setName	Void	Public	Name not null	Sets user name
setAddress	Void	Public	Address not Null	Sets user address
setPhone	Void	Public	Phone not null	Sets user phone
setEmail	Void	Public	Email not null	Sets user email
getName	String	Public	None	Returns user name
getAddress	String	Public	None	Returns user address
getPhone	String	Public	None	Returns user phone
getEmail	String	Public	None	Returns user email

Class Name: Theater				
Description: Contains information on a theater				
Attributes	Data Type	Access Level	Default	Description
TheaterName	String Length 1 ... 25	Private	"ABC Main Theater"	Holds the theater name

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

TheaterAddress	String Length 1 ... 25	Private	"123 ABC Place Kennesaw GA"	Holds the theater address
TheaterZip	String Length 5	Private	"30144"	Holds the zip code of a theater
TheaterPhone	String Length 10	Private	"000-000-0000"	Holds the theater phone number
Movies	List Dynamic List Max 100	Private	Initialize an empty list Run \$movieshowing = mysql_query(select all from MOVIE table where release date is not more than 21 days from today) for each in \$movieshowing add to list Return list	
Showtimes	List Dynamic List Max 100	Private	Initialize an empty list Run \$Showtime = mysql_query(select movieShowtime from MOVIE where showDate is TODAY) for each in \$Showtime add to list Return list	
TheaterDate	Date DateTime Format	Private	January 1, 1970	Holds the date to search within for movies playing in the theater
Function	Return Type	Access Level	Restrictions	Method Description
setTheaterName	Void	Public	Theater name not null	Sets theater name
setTheaterAddress	Void	Public	Theater address not null	Sets theater address
setTheaterZip	Void	Public	Theater zip not null	Sets theater zip
setTheaterPhone	Void	Public	Theater phone not null	Sets theater phone number
addMovie	Void	Public	Movie name not null	Adds movie to the theater
removeMovie	Void	Public	Movie name not null and must exist in collection	Removes movie from the theater

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

getMovies	List Dynamic List Max 100	Public	Initialize an empty list Run \$movie = mysql_query(select idMovie from MOVIE where movieName = "(user's choice)" for each in \$movie add to list Return list	
addShowtime	Void	Public	Run \$newShowtime= mysql_query(insert into MOVIE values '(showtime)' where movieID = 'xxxx')	
removeShowtime	Void	Public	Run \$newShowtime= mysql_query(delete MOVIE where movieID = 'xxxx')	
getShowtimes	List Dynamic List Max 100	Public	Initialize an empty list Run \$Showtime = mysql_query(select Showtimes from THEATER where movieName = '(user's choice)' for each in \$Showtime add to list Return list	
setDate	Void	Public	Date must be in format : January 1, 1970	Sets date for user to search within
getDate	Date DateTime Format	Public	None	Returns the date to search within
getTheaterName	String Length 1 ... 45	Public	None	Returns theater name
getTheaterAddress	String Length 1 ... 25	Public	None	Returns theater address
getTheaterZip	String Length 5	Public	None	Returns theater zip
getTheaterPhone	String Length 10	Public	None	Returns theater phone number

Class Name: Administrator				
Description: Defines administrator responsibilities and capabilities				
Attributes	Data Type	Access Level	Defaults	Description

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

AdminName	String Length 1 ... 45	Private	"John Do"	Holds the name of the Admin
AdminTitle	String Length 1 ... 45	Private	"Intern"	Holds the Admin title or position in the company
AdminPassword	String Length 6 ... 45	Private	"0000000"	Holds the admin password
AdminEmail	String Length 6 ... 45	Private	"admin@abc.com"	Holds the admin email
AdminEmployeeNumber	String Length 1 ... 45	Private	"-1"	Hold the admin employee ID number
Function	Data Type	Access Level	Restrictions	Method Description
setAdminname	Void	Public	Admin name not null	Set the admin name
setAdminTitle	Void	Public	Admin title not null	Set the admin Title
setAdminPassword	Void	Public	Admin password not null and must be six characters long	Set the admin password
setAdminEmail	Void	Public	Admin email not null	Set the admin email
setAdminEmployeeNumber	Void	Public	Employee number not null	Set the admin employee ID number
setDiscount	Void	Public	Discount not null	Set the discount rate or value
getAdminName	String Length 1 ... 45	Public	None	Return the admin Name
getAdmintitle	String Length 1 ... 45	Public	None	Return admin Title
getAdminPassword	String Length 6 ... 45	Public	None	Return the admin password
getAdminEmail	String Length 6 ... 45	Public	None	Return the admin Email

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

getAdminEmployeeNumber	String Length 1 ... 45	Public	None	Return the admin Employee ID number
setAgeLimit	Void	Public	Age limit must be > 18.	Sets the age limit for movie goers based on the MPAA rating
getComments	String Length 10 ... 45	Public	Comment Must exist	Returns the user comments of a movie
getUserRatings	String Length 6 ... 10	Public	User Rating must exist	Returns the user ratings
generateReports	String Length 10 ... 100	Public	Report Must exist	Returns the generalized report of a movie
generateSuggestion	String Length 10 ... 100	Public	None	Returns suggestions of movies based on user's previously watched movies, demographics and unwatched movies
createBulkEmail	String Length 6 ... 45	Public	Must be valid email address format	Return a bulk email to users about new movies showing or suggestions

Class Name: Movie				
Description: Contains information on a movie				
Attributes (fields)	Data Type	Access Level	Defaults	Description
MovieName	String Length 4 ... 25	Private	"ABC"	Holds the name of the movie
MovieRating	String Length 1	Private	"0"	Holds the MPAA rating of the movie
MovieLength	String Length 1 ... 5	Private	"90"	Holds the running time of the movie

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

MovieRelease	Date DateTime	Private	“January 1, 1970”	Holds the release date of the movie
MovieDescription	String Length 1 ... 100	Private	“Description not yet added”	Hold a brief description of the movie
Methods (operations)	Data Type	Access Level	Restrictions	Description
setMovieName	Void	Public	Movie name not null	Sets the name of the movie
setMovieRating	Void	Public	Movie rating not null	Sets the rating of the movie
setMovieLength	Void	Public	Movie length not null	Sets the running time of the movie
setMovieRelease	Void	Public	Movie release not null	Sets the release date of the movie
setMovieDescription	Void	Public	Movie description not null	Sets the description of the movie
getMovieName	String Length 1 ... 25	Private	Movie must exist	Returns the name of the movie
getMovieRating	String Length 1 ... 5	Private	Movie must exist	Returns the rating of the movie
getMovieLength	String Length 1 ... 5	Private	Movie must exist	Returns the running time of the movie
getMovieRelease	Date DateTime	Private	Movie release must exist and must be in correct DateTime format	Returns the release date of the movie
getMovieDescription	String Length 1 ... 100	Private	Movie description must exist	Returns the description of the movie
getTheater	String Length 1 ... 25	Private	Theater must exist	Returns a designated theater

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

Class Name: Comments				
Description: Contains of a movie				
Attributes	Data Type	Access Level	Default	Description
commentID	Integer 32 bit unsigned int	Private	0	Holds the auto generated comment ID
username	String Length 6 ... 25	Private	"Anonymous"	Holds the commenter's username or screenName
Comment	String Length 1 ... 100	Public	"Comment has not yet been added"	Holds the comment from end-user
Function	Return Type	Access Level	Restrictions	Description
setComment	Void	Public	Comment not null. Run \$newComment = mysql_query(insert into COMMENT (commentDescription) values ("(user comment)") where movieID = 'xxxx')	
editComment	Void	public	Comment not null and must already exist to be edited update COMMENT set commentDescription='new comment' where commentID = xxx;	
getComment	String Length 6 ... 100	Public	None	Returns the comment for a movie
getCommentID	Integer 32 bit unsigned int	Public	None	Return the auto-generated commentID

Class Name: Suggestions				
Description: Contains suggestions of a movie based on previously watched movies and demographics				
Attributes	Data Type	Access Level	Default	Description
WatchedMovies	List Dynamic List Max	Private	Empty list	Holds the names of the previously

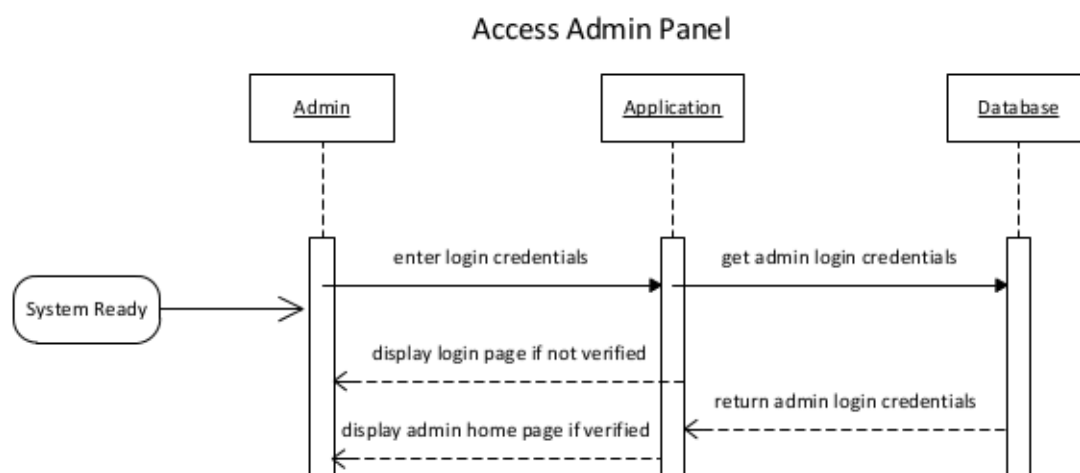
Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

	100			watched movies
NotYetWatchedMovies	List Dynamic List Max 100	Private	Empty list	Holds the name of movies of movies not yet watched
Demographics	String Length 5 ... 40	Private	“dems not yet added”	Holds the user’s demographic information
Function	Return Type	Access Level	Restrictions	Description
setMoviesWatched	Void	public	Movies watched not be null Run \$movieWatched = mysql_query(insert into USER (userMoviesWatched) values (“(movie watched)”))	
setMoviesNotYetWatched	Void	public	Movies not yet watched not null	Sets the movies not yet watched
setDemographics	Void	public	Demographics not null	Sets the user demographics
getMovieRating	String Length 5	Public	Movie Rating must exist	Returns the rating of the movie
getMoviesWatched	List Dynamic List Max 100	Public	Run \$moviesWatched = mysql_query(select userMoviesWatched from USER where userName = “Jane Doe”	
getMovieNotYetWatched	List Dynamic List Max 100	Public	None	Returns a List of not yet watched movies
getDemographics	String Length 5 ... 40	Public	None	Returns the user Demographics
getMovieRatings	String Length 5	Public	Run \$movieRating = mysql_query(select movieRating from MOVIE where MovieName = ‘xxxxxxx’;	
getMovieGenre	String Length 5... 10	Public	None	Returns the movie
getMostLikesMovies	List Dynamic List Max 100	Public	None	Return a list of user’s most liked movies or movies with good ratings

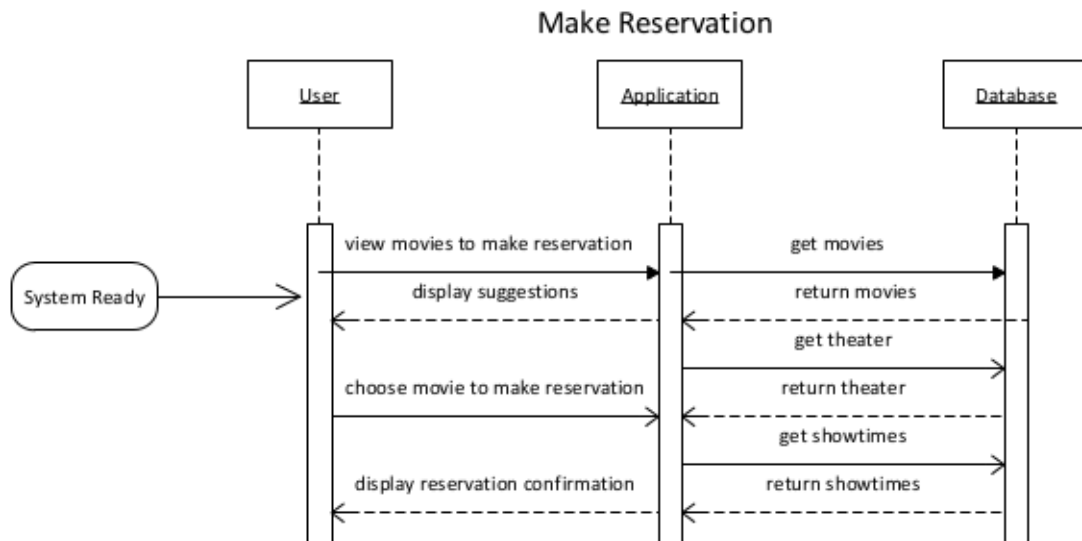
Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

				that the user has not watched yet
--	--	--	--	-----------------------------------

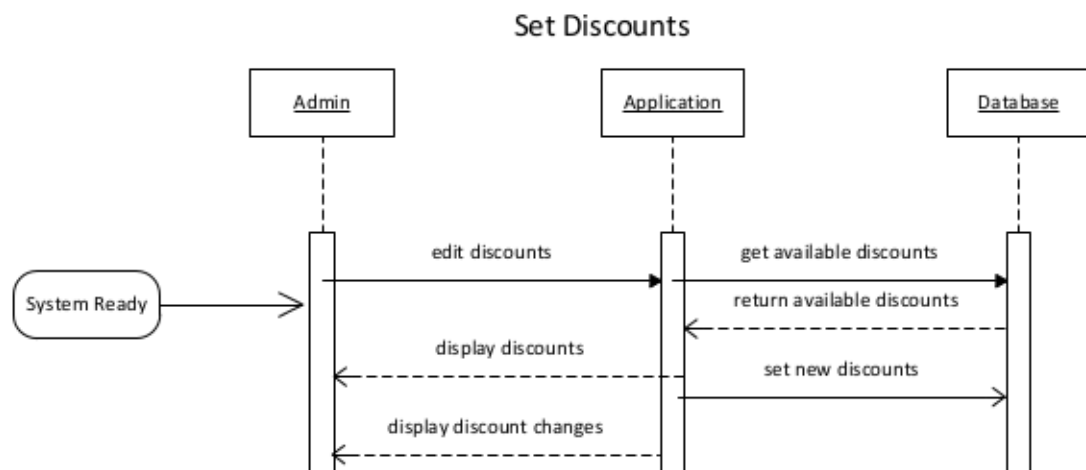
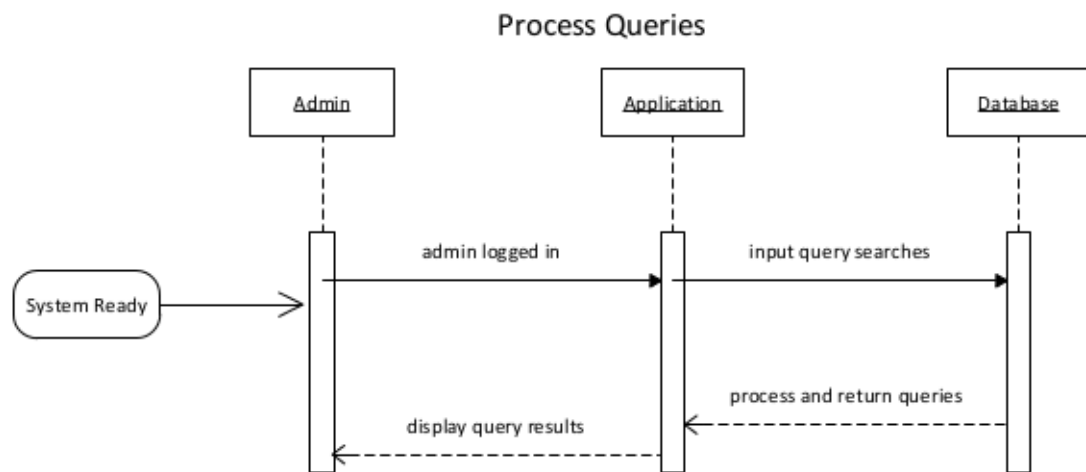
3. Sequence Diagrams



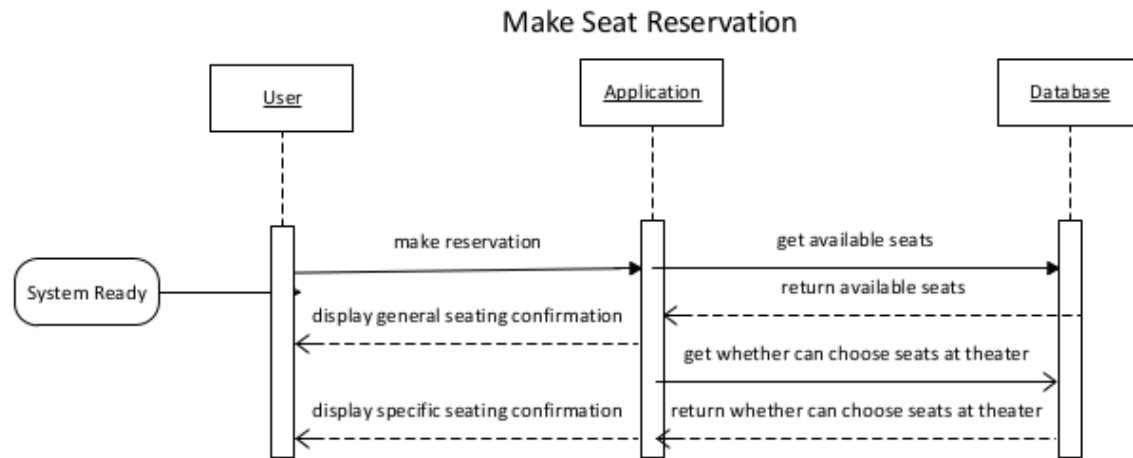
Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	



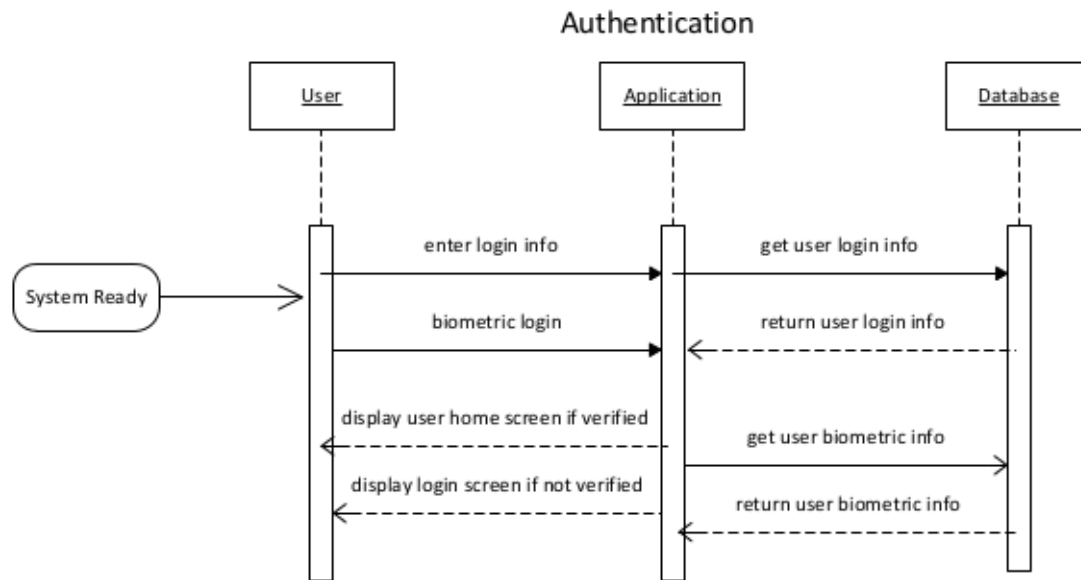
Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	



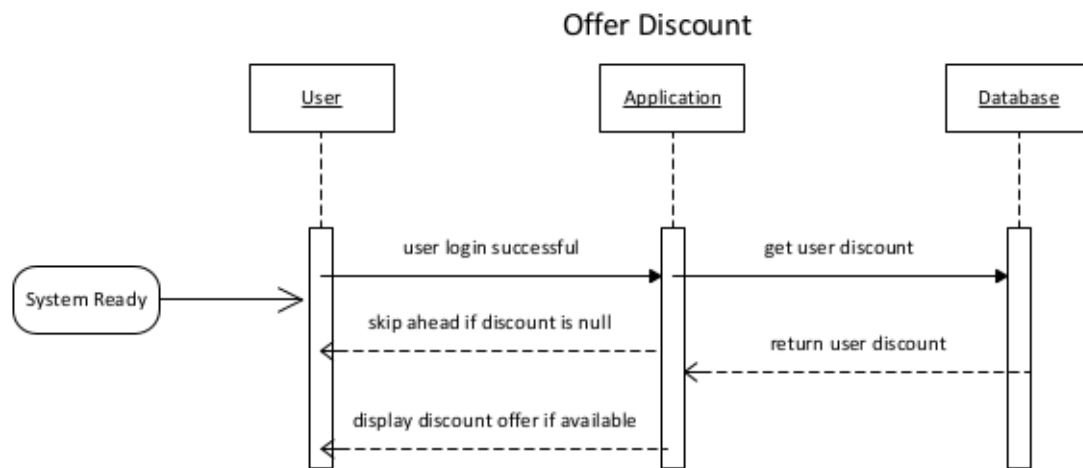
Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	



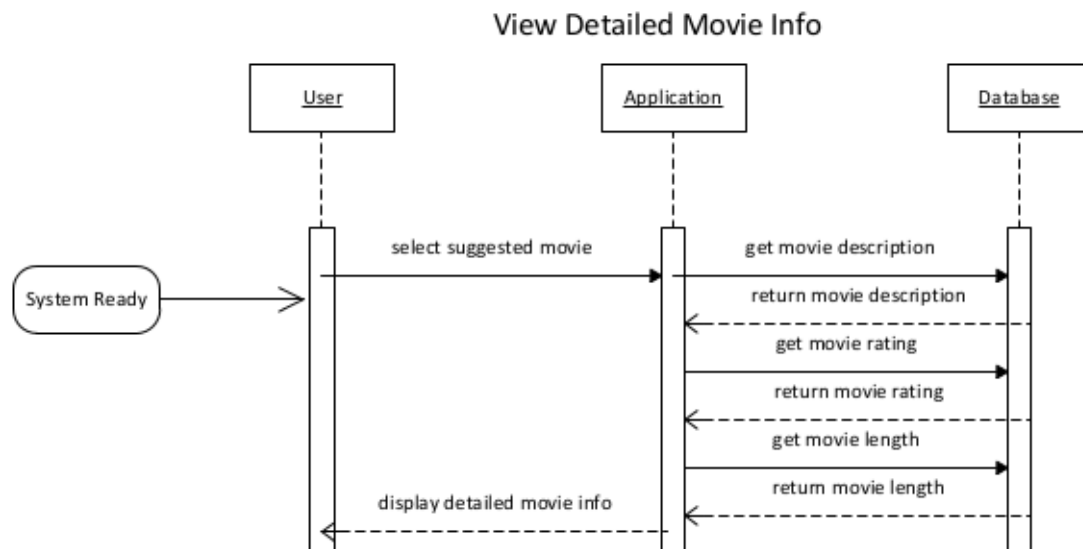
Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	



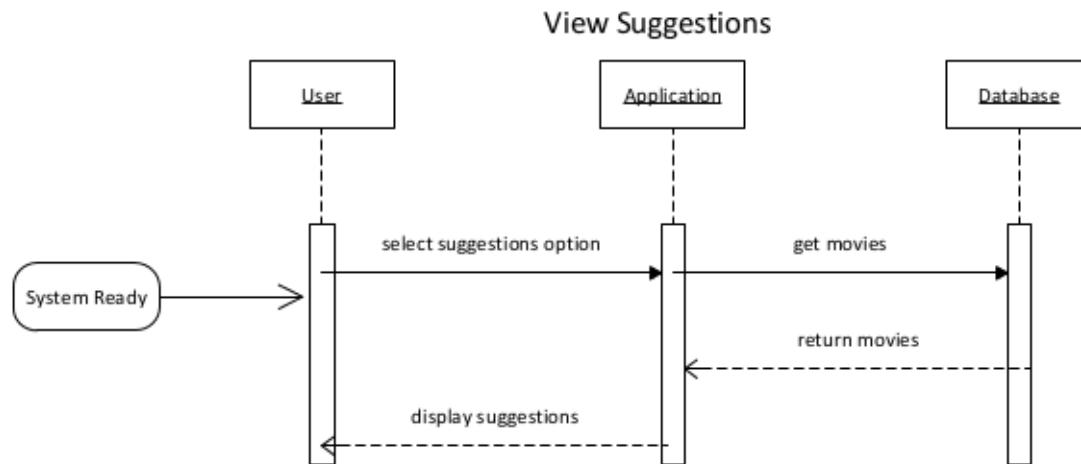
Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	



Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

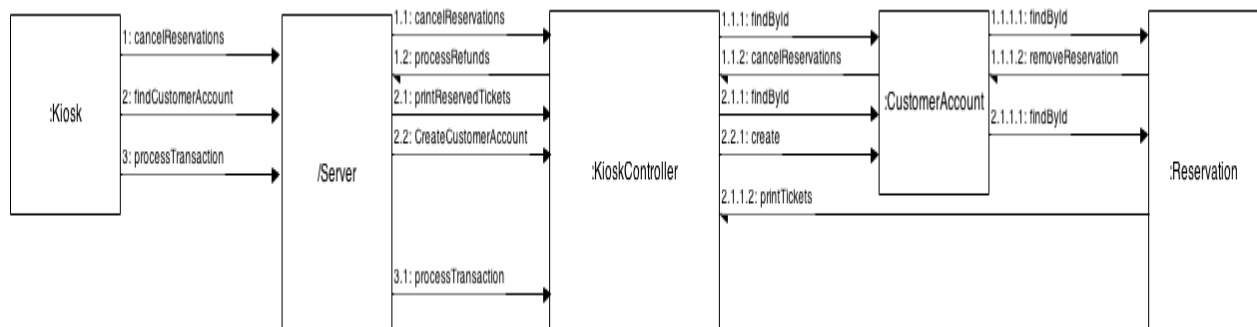


Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	



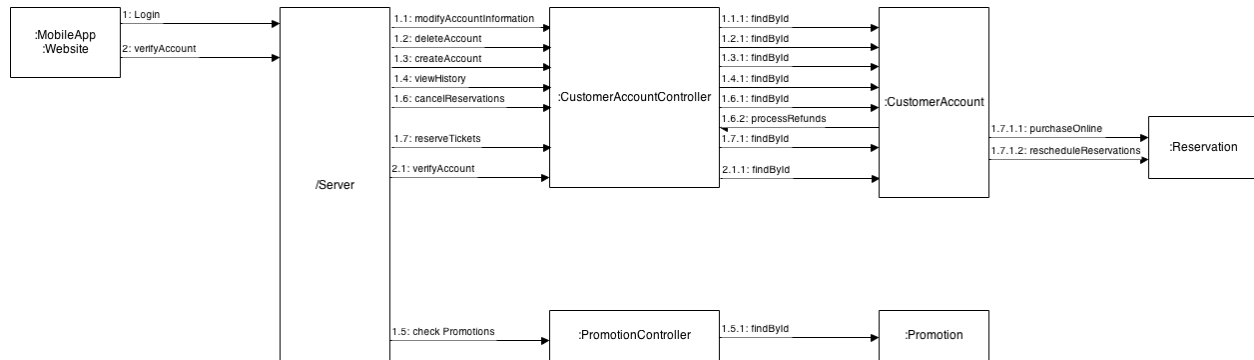
4. Object Collaboration

4.1 Kiosk

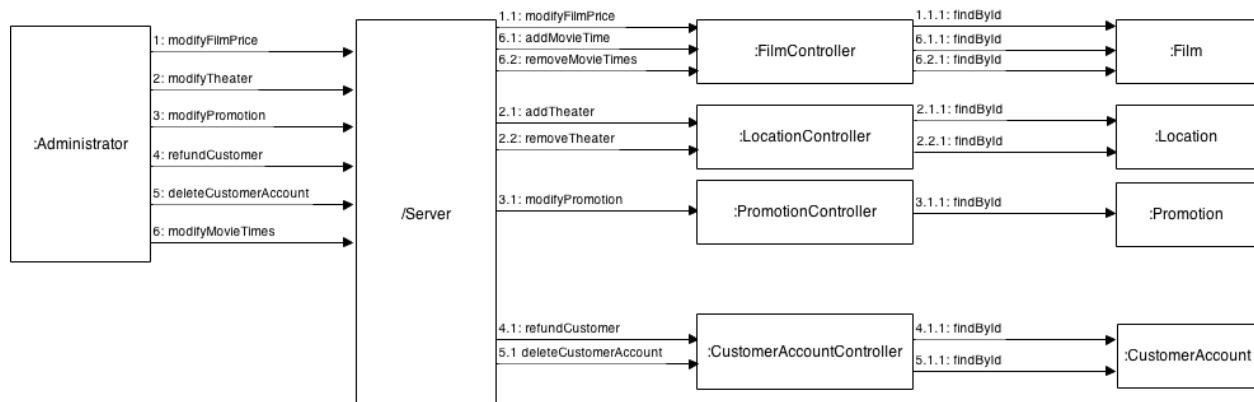


Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

4.2 Mobile app/ Website

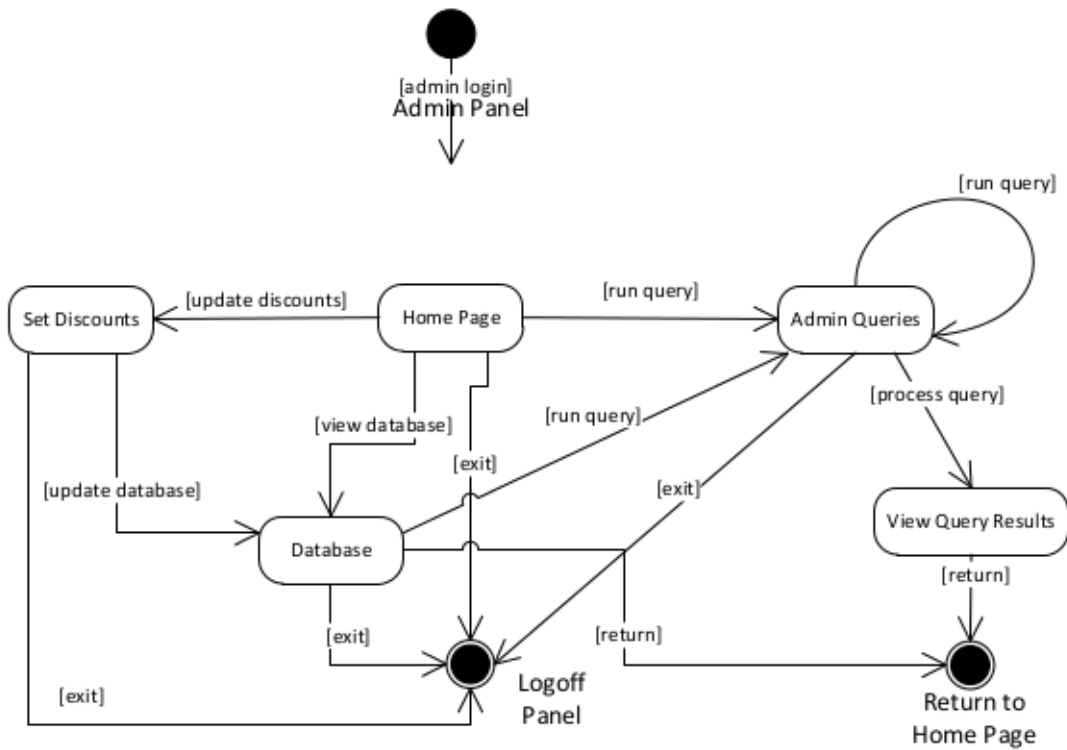


4.3 Administrator

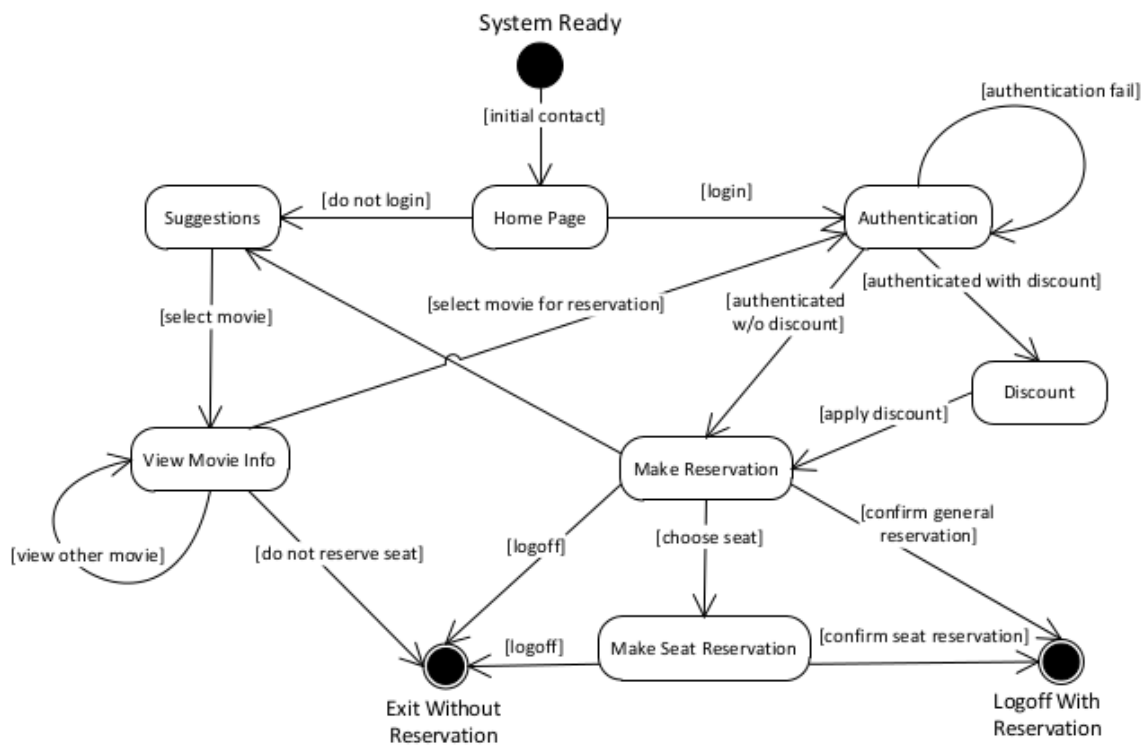


Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

5. Object behavior



Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	



6. Data Design and Database Design

6.1 Internal Software Data Structure

ABC's internal structure is divided into two parts: server-side and client-side. On the client side, data will reside locally in memory and will be organized based on the classes defined earlier in this document. Since the ABC reservation program may be considered data-centric, the classes that handle the data will be isolated and will be accessed by way of a model view controller system. The data on the local mobile client / kiosk will be requested from the server at application initialization and refreshed as necessary based on user actions. The data structure on the server will essentially mirror the structure of the local Android client in terms of member fields of the classes. The server will be implemented using PHP. Permanent storage of user information will be accomplished using a

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

MySQL database. Section 3.8 (Server Component) covers this subject in further detail. The server and mobile client will exchange data using the JSON format. JSON is a lightweight object description language that is similar to XML. JSON is being used due to its versatility and because of the fact that an implementation is available for both PHP and JAVA.

6.2 Global Data Structure

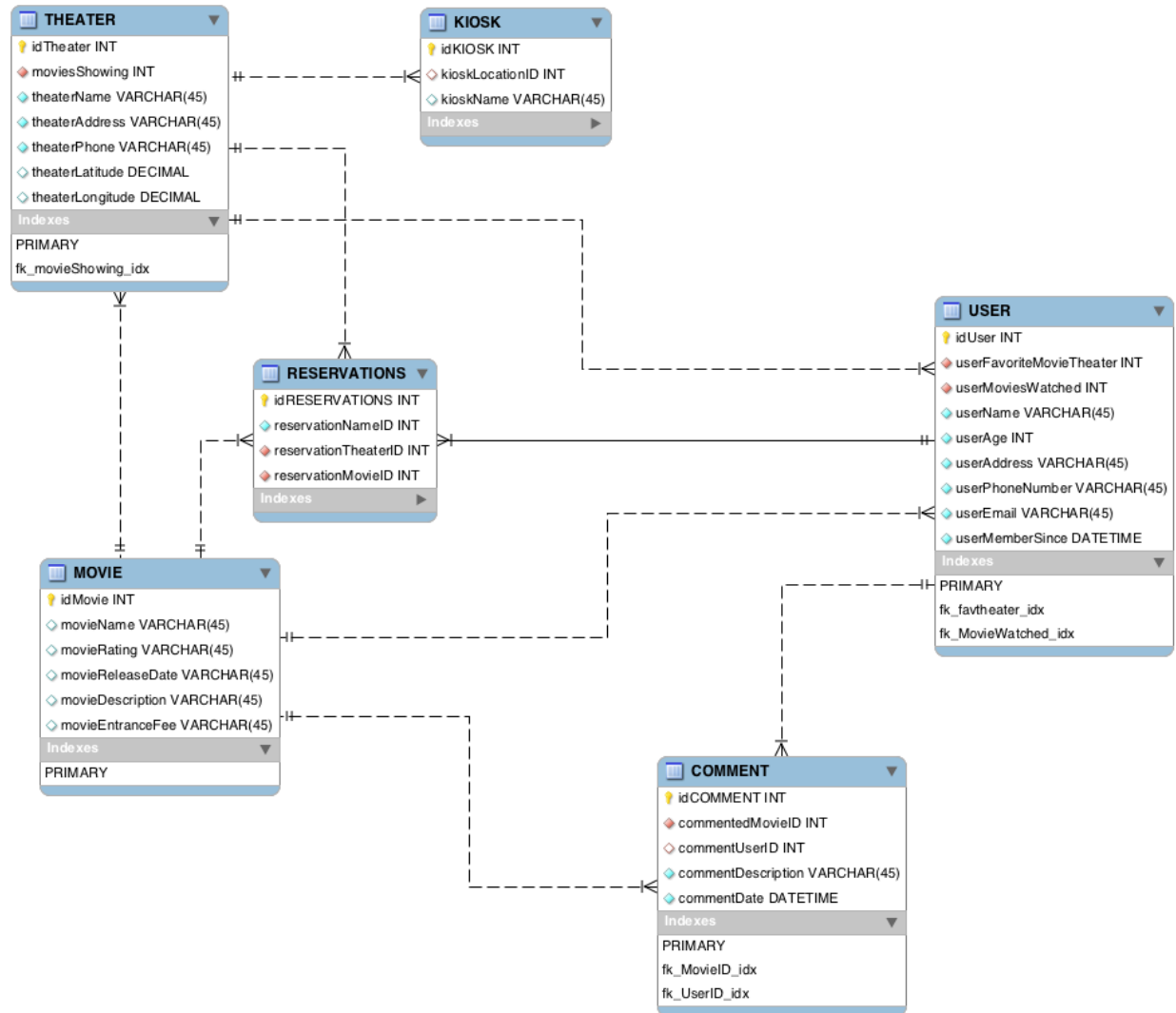
The global data structure of this application is best characterized by the database. The database structure shows the data involved in the application in its purest sense. The local mobile client of ABC will never access this database directly; it will instead issue requests to the server.

6.3 Temporary Data Structure

Temporary data structures, as they relate to ABC, refer to the data objects that are created on the local Android client, and also to the JSON objects that interchanged between the server and the client. The data objects created on the local device will only exist for the duration of time that the application is running, and will subsequently be destroyed. The JSON objects will only exist for the duration of the transaction between the client and server. The server will destroy the objects after sending them, and the client will destroy the JSON objects once they have been parsed.

6.4 EER Database Design

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	



7. Other Design Requirements

7.1 Help Menu

- Displays a list of topics covering the different components of ABC Theater reservation System
- Offers detailed information on each feature, menu, etc.
- Can be accessed at any time via the Settings menu

7.2 Settings Menu

- Allow user to customize his/her preference
- Enables the user to modify certain features and functionalities

Theater Reservation System	Version: 1.0
Software Design Document	Date: 28-MAR-14
SDD-Doc-1	

- Can be accessed at any time using the built-in Settings button on Android phones

7.3 PayPal Integration

- Incorporates a mechanism for initiating real transactions
- Facilitates secure, hassle-free transactions between members
- Automatically updates member balances as transactions occur

7.4. GPS Tracking

- Stores location data associated with certain events
- Utilizes Google Maps to display transaction locations
- Creates an expense map which can be viewed by all members of a group

7.5. Receipt Imaging

- Utilizes the camera built into mobile devices
- Records and stores a snapshot of receipts associated with different expenses
- Provides a method of checking/verifying expenses posted to a group

7.6. E-Mail/SMS Notifications

- Extends the standard notifications service built into the software
- Automatically delivers notifications via e-mail and/or text message
- Enables individuals without accounts to receive group notifications

7.7. Tutorial

- Provides an abridged version of the Help menu for first-time users
- Offers a step-by-step run through of each feature, menu, etc.
- Enables any user to quickly and easily take advantage of all the reservation system's functionalities