Project Group Members (Course Group 13):

Poojah Karunakaran (100618754)
Kevin Erskine (100762922)
Ivan Bisol (100701735)
Misha Larionov (100743093)

Architectural Concerns:

ID	Concern
CRN-1	Establishing an overall initial system structure.
CRN-2	Leverage the knowledge of Java from the members of the development team to create the application
CRN-3	Allocate work to members of the development team

ADD Step 1: Review Inputs

UC-1: Account creation	A user will create their account using their email and a password following certain security guidelines. The user's login information will then be stored in a secured database. The user will then fill out additional information such as name, preferred vehicle, contact information, and address.
UC-2: Location selection	Users will have their "home" theatre auto selected based on their location when first creating an account with the option to select a different theatre if desired.
UC-3: Movie selection	As users go to search for tickets, their selected theatre is queried to display all movies showing for the next week by default and users can change the desired dates.
UC-4: Time slot selection	Once users select a certain movie to view, they are then shown all the available time slots for that movie at their specific theatre.

Use Cases for reference

Category	Details
Design Purpose	The purpose is to create an appealing design for users and a design that is able to update details consistently and promptly .
Primary Functional Requirements	From the use cases presented, the primary ones are determined to be: UC-1: Necessary to the use of service provided by website UC-2: Necessary to the use of service provided by website UC-3: Necessary to the use of service provided by website UC-4: Necessary to the use of service provided by website

Quality Attribute Scenarios:

Scenario ID	Importance to Customer	Difficulty of Implementation according to the Architect
QA-1	**High**	**Medium**
QA-2	**High**	**Medium**
QA-3	**High**	**High**
QA-4	**Medium**	**High**
QA-5	**Medium**	**High**
QA-6	**Medium**	**Low**

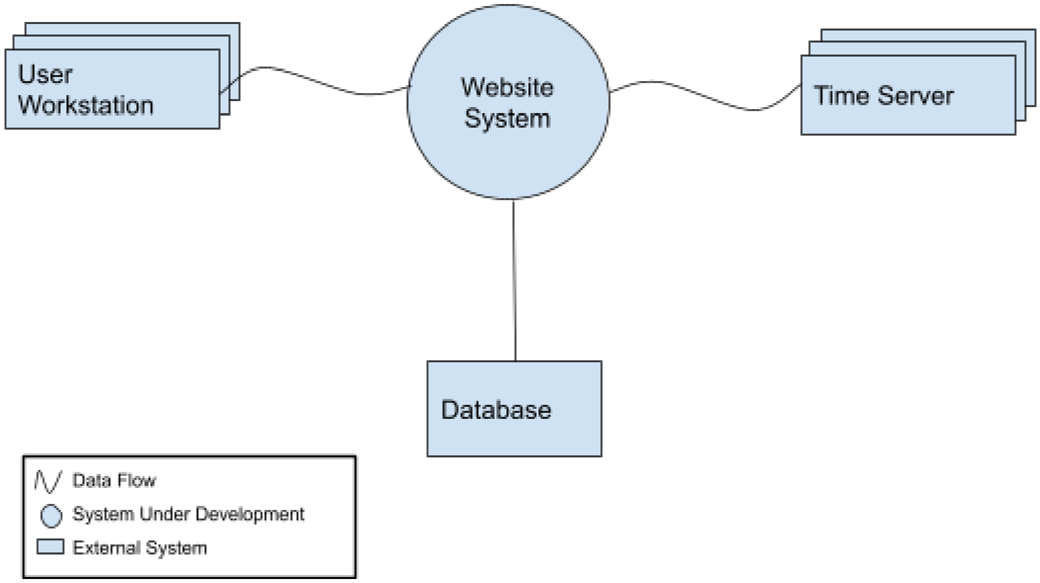
From this list QA-1, QA-2 and QA-3 are selected as drivers.

Constraints : All constraints are included as drivers.
Architectural concerns: All architectural concerns are included as drivers.

Step 2 : Establish Iteration Goal by Selecting Drivers

This step will illustrate the general concern of establishing the overall system structure of the website that is being created.

This is created while being mindful of the following:
QA-1: Availability
QA-2: Usability
QA-3: Security and Privacy
CON-1: The system must be accessible and functional on all major browsers (Chrome, Firefox) on all desktop platforms (Windows, OSX, Linux)
CON-2: The system must store user data securely and passwords must not be visible, even in the event of a data breach
CON-3: The system must be able to handle a minimum of 25 simultaneous users
CON-4: Ticket information must be stored and accessible by user for 30 days after movie is over
CON-5: All collected information must be stored and accessed from a database



Context Diagram 1.1

Step 3: Choose One or More Elements of the System to Refine

The main system and the system that is accessed by the user needs to be refined. We would have to break down the system by first working on the system on a website before working on the system accessed by the user.

Step 4 : Choose One or More Design Concepts That Satisfy the Selected Drivers

Design Decisions and Location	**Rationale**
Logically structure the client part of the system using the **Web Application** reference architecture.	The Web Application reference architecture allows for easy deployment across all modern browsers and desktop platforms. Since there are no requirements for rich user interaction, the Web Application reference is utilized.
Microservice Architecture	This Architecture is meant to isolate the layers which are needed for a system such as this where there are multiple transactions and updates. This Architecture caters towards automated transactions and when the system has to be changed, it can be done with the IT responding to those changes.
Physically Structure the application using the **three-tier deployment pattern**	The system must be accessed from a web browser and an existing database. Therefore, a three-tier deployment pattern would be the most appropriate for this situation.
Build user interface of client application using Java Swing framework and other Java technologies	Java rich clients ensure portability. Architects are familiar with Java. Therefore, this would be the most appropriate choice at this location.

Discarded Alternatives:

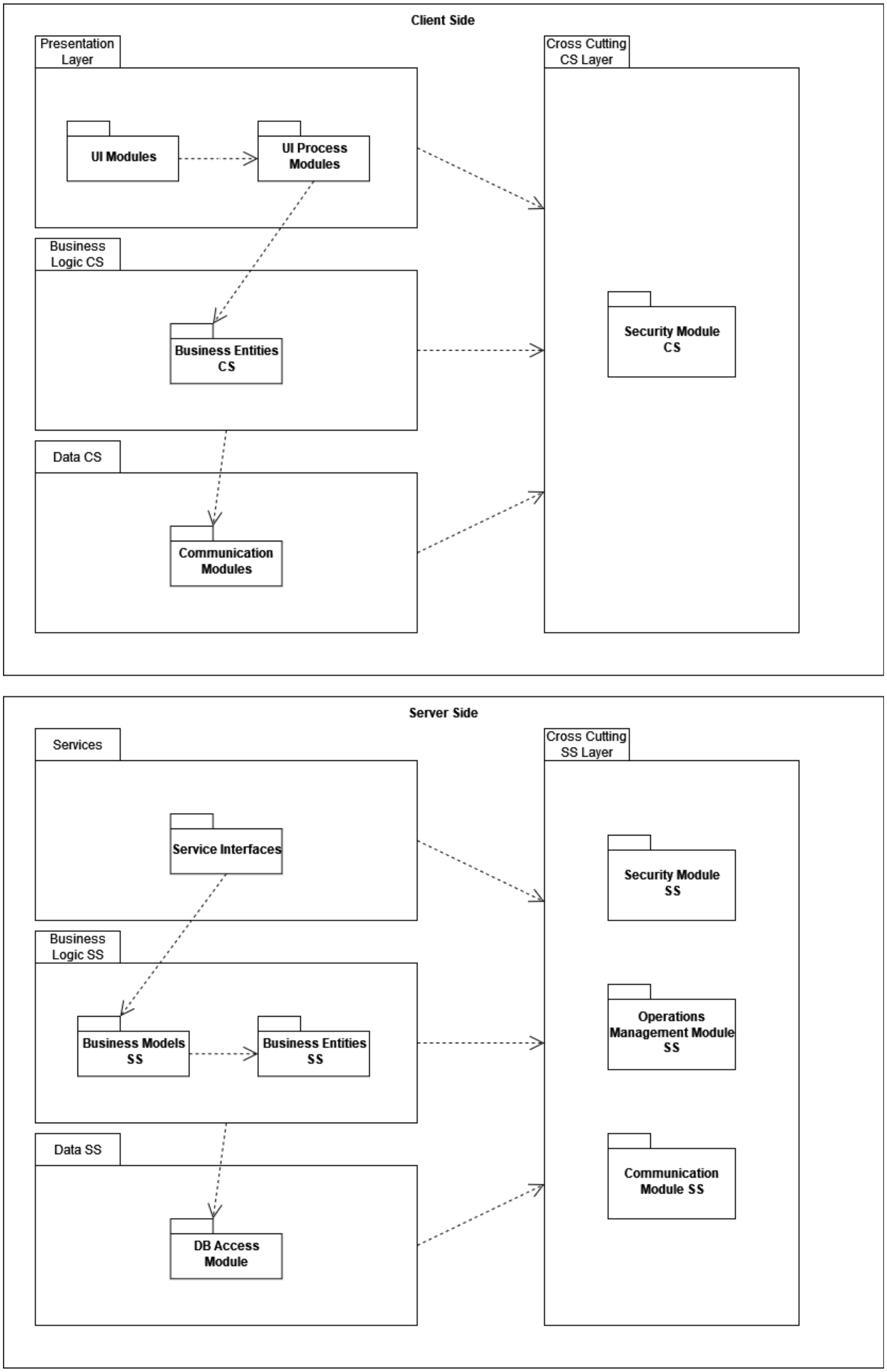
Mobile Applications	This reference architecture is oriented towards the development of applications that are deployed on handheld devices. The application being developed does not require to be accessible on handheld devices. This can be later developed and implemented but it is not need for this specific application
Rich Client Application	Rich Client Applications are more intensive then we'd need for the deployment of our software. Additionally. having to create a standalone client for various desktop platforms can introduce

	additional errors.

Step 5 : Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

Design Decision	Rationale
Create a module dedicated to accessing the database	This module will service to accessing and storing data in the database
Remove local data sources in the Web Application.	To ensure accurate and consistent information is displayed at all times, data will entirely be sourced by the database to ensure all users see the same ticket availability.

Step 6 : Sketch Views and Record Design Decisions



1.2 Module View Diagram

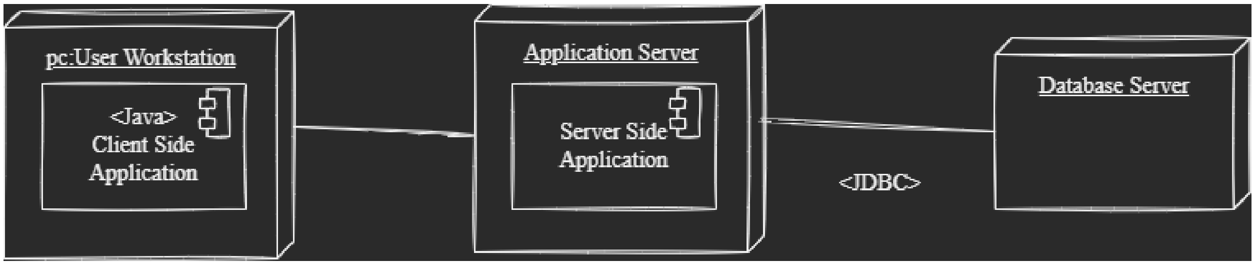
Element	**Responsibility**
Presentation Layer	Modules in this layer take care of user interaction use case control flow
Business Logic CS	These module perform business logic operations that are executed on the client side
Data CS	Modules responsible for communication
Cross-cutting CS	This layer includes modules with functionality that goes across different layers. This is helpful for to achieve the QAs specified
UI Modules	These modules are responsible for receiving user input and creating the user interface
UI Process Modules	Controls flow of all system use cases
Business Modules CS	These implement business operations that are performed locally
Business Entities CS	These make up the domain module
Communications Modules CS	These modules are focused on taking the services provided by website application on the server side
Services Server Side (SS)	Modules are focused on exposing services that are consumed by clients
Business Logic SS	Business logic operations that are required are performed by the modules in this layer
Data SS	Modules communicates with the time server
Cross-Cutting SS	These modules goes across to different layers
Server Interfaces SS	Modules expose services that are consumed by the clients
Business Modules SS	The responsibility of these modules is to implement business operations
Business Entities SS	Responsible in the makeup of the domain model
DB Access Module	Module is responsible to access the database and input and receive information from said database
Time Server Access Module	Module is responsible to communicate with time server and provide appropriate information when requested

The responsibilities of elements summarized:

Element	**Responsibility**
User Workstation	The user's PC host the client side logic of the application
Website Server	Displays the webpage
Database	Stores all the appropriate information for access
Time Server	Keeps track of time

Table summarizing the relationships between elements in the diagram:

Relationship	**Description**
Between web/app server and database server	Communication between these three elements are achieved using the JDBC protocol



1.3 Initial Deployment Diagram for the drive-in theatre website system

Step 7: Perform Analysis of Current Design and Review Iteration

Not Addressed	**Partially Addressed**	**Completely Addressed**	**Design Decisions Made During the Iteration**
	UC-1		Selected reference architecture establishes the modules that will support this functionality
	UC-2		Selected reference architecture establishes the modules that will support this functionality
	UC-3		Selected reference architecture establishes the modules that will support this functionality
	UC-4		Selected reference architecture establishes the modules that will support this functionality
		QA-1	The use of the Web Application Reference Architecture helps to ensure that the site maintains availability
QA-2			No relevant decisions made. As the requirements to

			meet in usability will be addressed in later iterations
	QA-3		The use of Web Applications helps to ensure the security and privacy required by the system. This will be addressed in greater detail in later iterations
		CON-1	The use of Java technologies allows for this system to be supported on all major browsers and desktop platforms
	CON-2		The use of Web Applications helps to ensure the security and privacy required by the system. This will be addressed in greater detail in later iterations
		CON-4	The use of a database in the backend allows for information to be stored appropriately
		CON-5	The use of a database in the backend allows for information to be stored and accessed appropriately
		CRN-1	The appropriate architecture for this

			system has been chosen in this iteration
		CRN-2	The appropriate technologies that will be used in the creation of this system has been chosen is compatible with the knowledge of the developers
CRN-1			No relevant decisions have been made