

Iteration 1: Establishing an Overall System Structure

In this section the results of the design process activities performed in the steps of the ADD method first iteration are represented.

Step 2: Establish Iteration Goal by Selecting Drivers

This is the primary iteration of the design of a brownfield system Game_Knight, the iteration goal is tied to the first architectural concern CRN-1 *The initial structure of the system will determine the usability of the system* producing an initial overall system structure.

Along with the concern the architect must factor in all the drivers that influence the structure of the Game_Knight system. Below is a list of drivers the architecture will be required to be conscious of:

- QA-1: Availability
- QA-2: Modifiability
- QA-3: Performance
- QA-6: Usability
- CON-1: A catalogue of games must already be downloaded to utilize the functions within the system
- CON-2: A pre-existing database must be used to fetch the info for the games
- CON-3: A stable internet connection must be available
- CON-4: Performance information must be collected and logged for a minimum of 30 days
- CRN-2: The background knowledge of the team is essential in determining whether the system can be built or not. SQL knowledge is necessary, python is necessary in implementing the views for the statistical analysis function

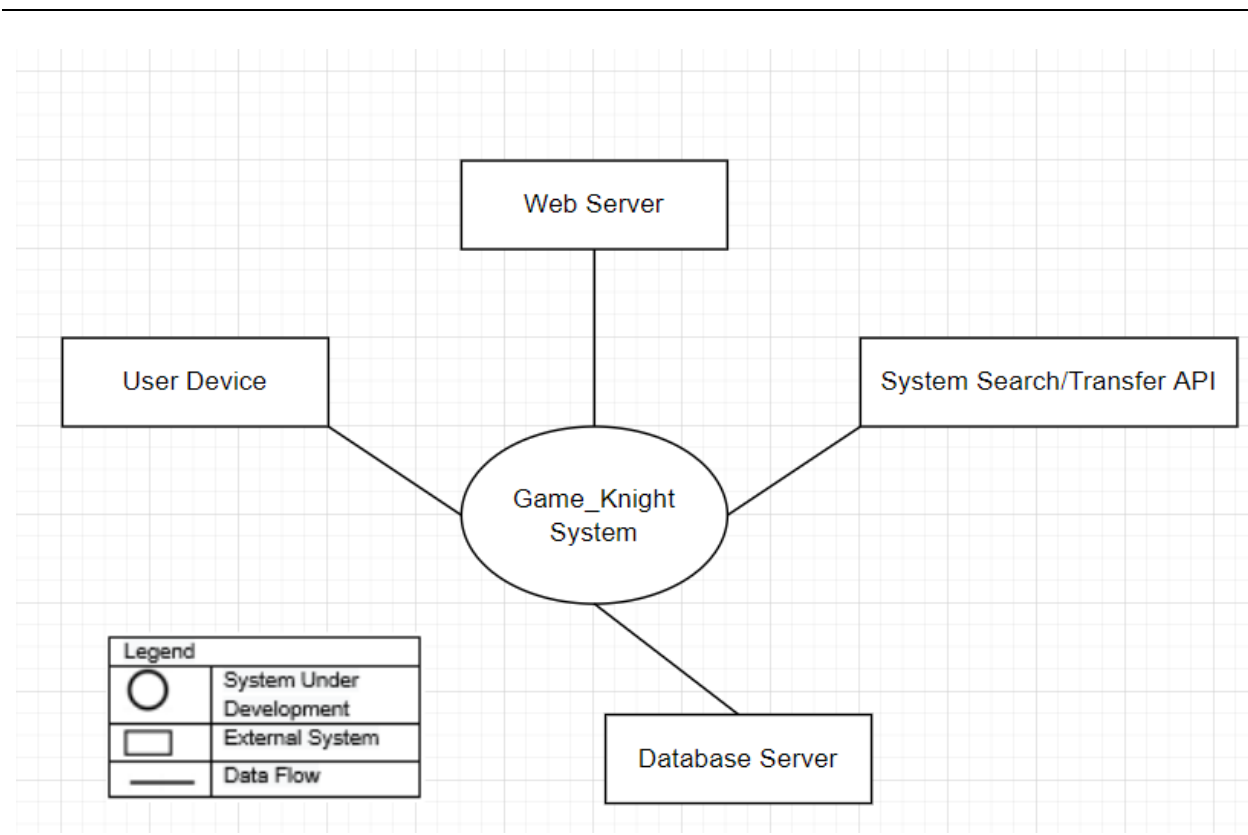


FIGURE 1.2 Context diagram for the Game_Knight system

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

This is a brownfield system, so this system has a previous build which heavily relies on the MYSQL Workbench database. To refine the system, it is necessary to refine the input stream of the Game_Knight system, which will be done using the search and transfer API.

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

In the primary iteration the goal of establishing an initial structure of the Game_Knight system design concepts are introduced. The table below briefs the selection of design decisions.

Design Decisions and Location	Rationale
Logically structure the client view of the system using a Web application	This reference architecture is used toward the development of applications that is typically initiated by the user from a web browser like google that communicates with a server using HTTP protocol. Since the bulk of the system resides in the MYSQL workbench database application

Discarded alternatives:	
Alternative	Reason for Discarding
Rich Internet applications	RIA typically run inside a browser which is great for the system and may be developed using code that is executed by the browser like Asynchronous JavaScript or XML (AJAX). Though this reference architecture is not needed as a much simpler approach
Mobile applications	This reference architecture is like a web application in terms of linking the system to a device, but it is deployed to handheld devices. This reference architecture was not used due to the fact it was not the greatest option when considering tying the database server.
Rich Client applications	This reference architecture is used typically when want an application to be highly interactive and responsible and desire to leverage the user's machine resources like a graphics card. This architecture wasn't used because there is no need to be highly interactive and responsible and desire to leverage the user's machine resources
Build the user functions using python and Django	The systems main functionality needs to be imbedded into the system using python functions will be built using the classes that are simplified from the database from the Django
Deploy the web application using html and API	Access to the application is obtained via a git hosted web browser, which launches the system This technology makes it, so the end-user does not have to download any applications.

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

The instantiation design decisions are explained in the table below:

Design Decisions and Location	Rationale
Web application to disconnect local sources of data	With a strong and stable connection, it is viable to not store any data locally. The data layer is where server communication is handled. Communication between the many client components are directed via local method calls without any other support.
Create a module dedicated to accessing the views pre-built in the database server.	The module will allow permission to view and close different iterations of the main page of the web application. Which has ties to QA-3 (performance), QA-6 (usability). This will further the development of the system's views as many different variants of existing pages in the site will be constructable.

What was examined from these instantiation results are elaborated on the next step of the iteration, though it is too vague in terms of functionality as this is the primary iteration. These results will be used in further iterations to make definitions more concise.

Step 6: Sketch Views and Record Design Decisions

The Figure 1.3 shows the module view of the reference architecture web application that is used in the client and server side.

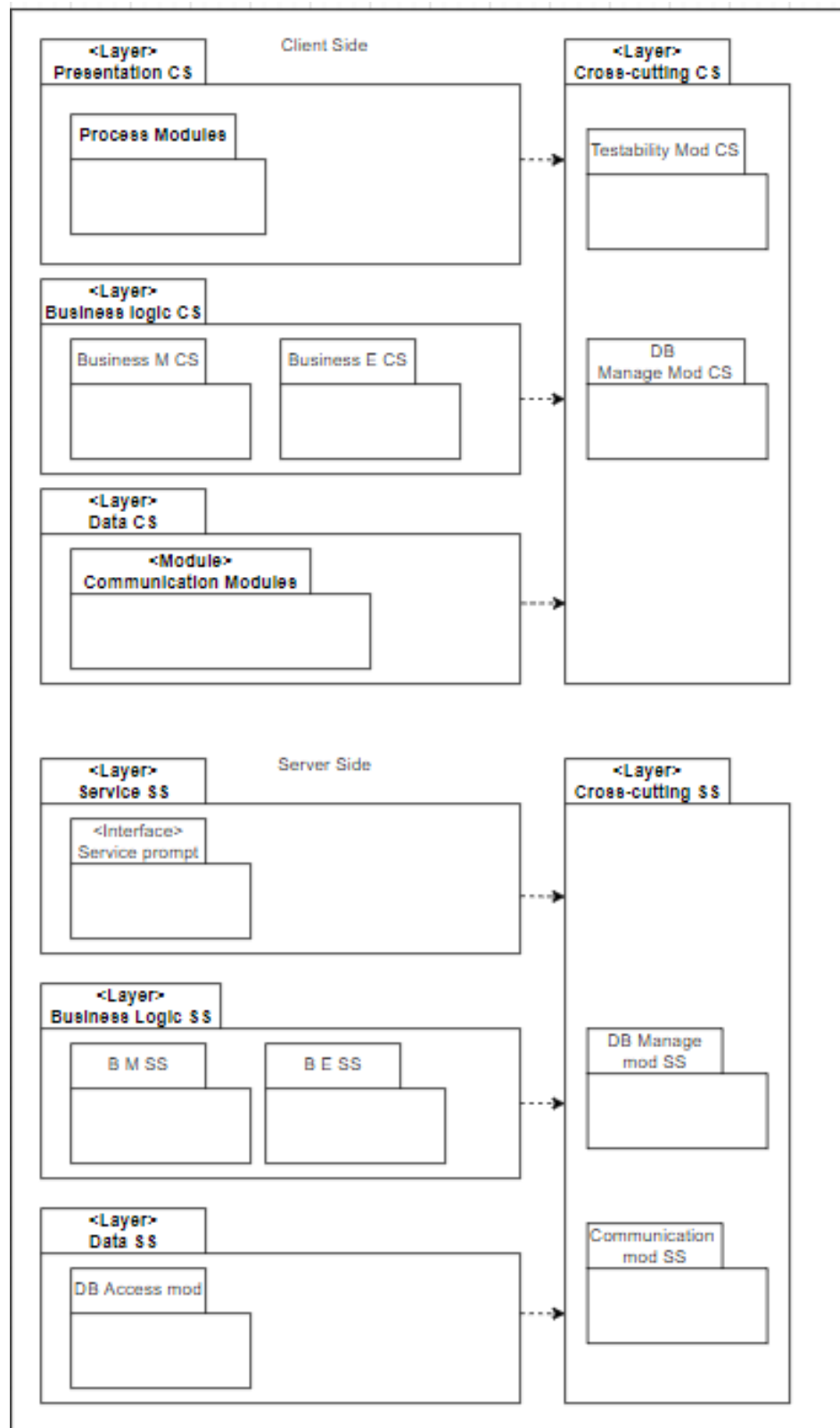


FIGURE 1.3

For Figure 1.3 a table is presented below naming each element and providing a brief description of the responsibility the element holds. As this is the primary iteration these descriptions are rough and unfinished.

Element	Responsibility
Presentation client side (CS)	This layer is used to show modules involved in the client side of the system
Business logic (CS)	This layer contains modules associated with the business functions in the system
Data (CS)	This layer discusses modules linked to the communication of the system
Cross cutting (CS)	This layer has modules involved in multiple layers of the client side of the system
Process module	This module is responsible for control flow in the system
Business modules (CS)	Show business functions from server side
Business entities (CS)	Make up the domain model
Communication module (CS)	Represents the communication on client side of the system
Services server side (SS)	Represents the services included on the server side of the system
Business logic (SS)	This layer contains modules associated with the business functions in the system
Data (SS)	This layer discusses modules linked to the communication of the system
Cross cutting (SS)	This layer has modules involved in multiple layers of the server side of the system
Service interface (SS)	
Business modules (SS)	Implement functions
Business entities (SS)	Make up the domain model
DB Access module	This module is responsible for linking the database to the system

The responsibilities of the elements concerned with the process of using the system are below:

Element	Responsibility
User device	The user's device that allows them to view the web application on a web browser
Web server	The server that host the application for the client-side view
Database server	The server in which the data for the tables are stored along with views of the table
API Application	The application that digs information of game data for the Game_Knight system

The initial deployment diagram is featured below:

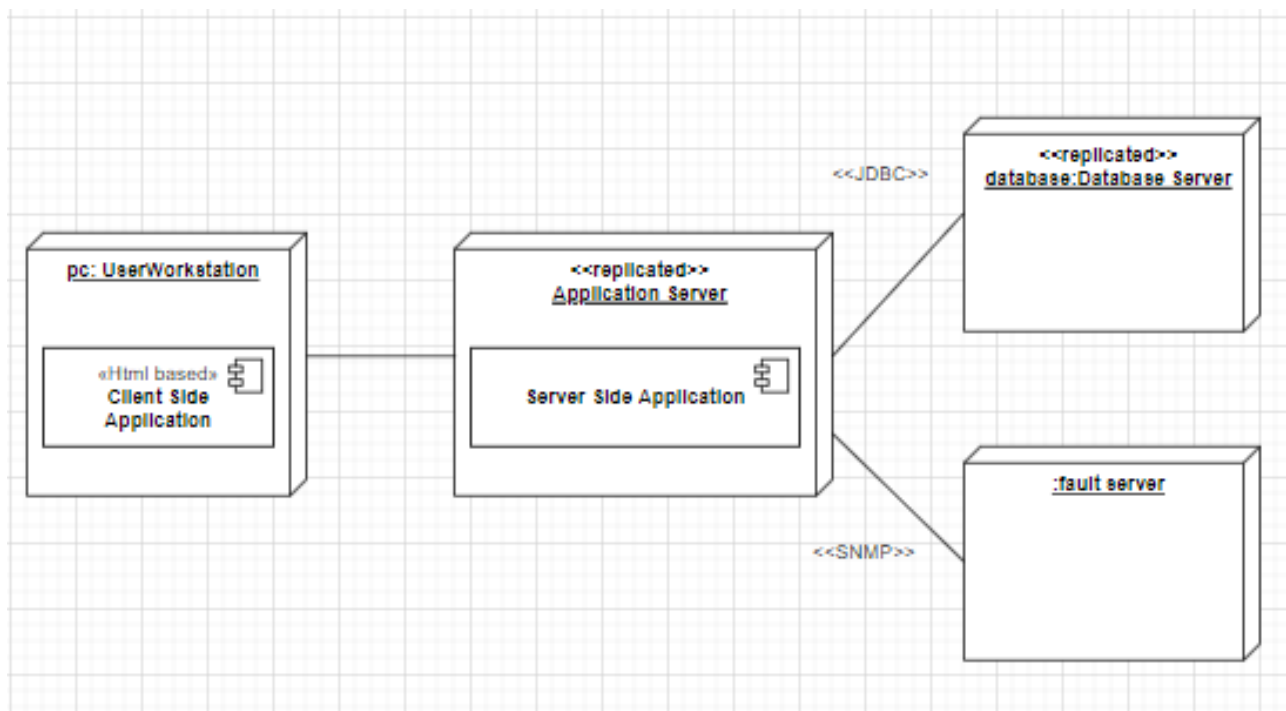


FIGURE 1.4 Initial deployment diagram

Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

A summary of the design process is explained in the table below

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During the Iteration
	UC-1		The web application reference architecture make it simple to view the clients view of the systems. Flags are set to further elaborate on the systems performance.
	UC-3		The selected architecture defines the modules that will support this case.
	UC-7		The selected architecture defines the modules that will support this case.
	QA-1		The Availability of the system would be 24/7 as it is a website
	QA-2		The modifiability of the system should be flexible as many new functions will be added when the database is extended though as it is too early to make any final decisions.
	QA-3		The performance of the system is crucial functions must return a value whether it completes the operation or it does not.
	QA-4		The security of the website is not a priority as the user is saving their information locally
QA-5			No Relevant decisions made
	QA-6		Data logging using the save tactic will aid in the usability attribute of the system.
		CON-1	A catalogue of games must already be downloaded to utilize the functions within the system this is taken care of as it is a brownfield system

CON-2		No Relevant decisions made
CON-3		No Relevant decisions made
CON-4		Use of the save tactic to manage data logging will aid in this constraint as it is required in the greater implementation.
CRN-1		Changes to the web application were made based on the concern
	CRN-2	All modules related to the use cases have been examined and identified and a matrix for the development distributed was created.
	CRN-3	No Relevant decisions made as this is in the user's hands.
CRN-4		No Relevant decisions made
