Step 1: Review Inputs

<table>
<tr><th>Category</th><th>Details</th></tr>
<tr><td>Design purpose</td><td>This is a Social Networking website. The purpose of this is to produce sufficiently detailed design to support the functionality of the website.
The system should provide the user the ability to create an account, login into their account, search for different users in the database or different posts in the system, add/remove friends, share/delete posts, modify their profile, etc.</td></tr>
<tr><td>Primary Functional requirements</td><td>Of the use cases presented in Figure 1.1.1, the primary ones are:
- UC -1 because it is the first step in accessing the website
- UC - 2 because it directly supports the core functionality
- UC - 6 because it directly supports the core functionality</td></tr>
<tr><td>Quality attribute scenarios</td><td>The scenarios which are explained above are prioritized in the following table:
<table>
<tr><th>Scenario ID</th><th>Importance to Customer</th><th>Difficulty of Implementation according to the architect</th></tr>
<tr><td>QA-1</td><td>High</td><td>High</td></tr>
<tr><td>QA-2</td><td>High</td><td>High</td></tr>
<tr><td>QA-3</td><td>High</td><td>Low</td></tr>
<tr><td>QA-4</td><td>Medium</td><td>Medium</td></tr>
<tr><td>QA-5</td><td>High</td><td>Low</td></tr>
<tr><td>QA-6</td><td>Medium</td><td>High</td></tr>
<tr><td>QA-7</td><td>High</td><td>Medium</td></tr>
</table>
</td></tr>
<tr><td>Constraints</td><td>All constraints are included as drivers.</td></tr>
<tr><td>Architectural Concerns</td><td>
<table>
<tr><th>ID</th><th>Concern</th></tr>
<tr><td>CRN-1</td><td>Accomplish completion of the final system in the allotted time and resources.</td></tr>
<tr><td>CRN-2</td><td>Evenly distribute the workload within the project across the development team to fully utilize the team's capabilities.</td></tr>
<tr><td>CRN-3</td><td>Creating the first prototype model of the system’s structure.</td></tr>
</table>
CRN-2 was selected as the driver.</td></tr>
</table>

Step 2: Establish Iteration Goal by Selecting Drivers

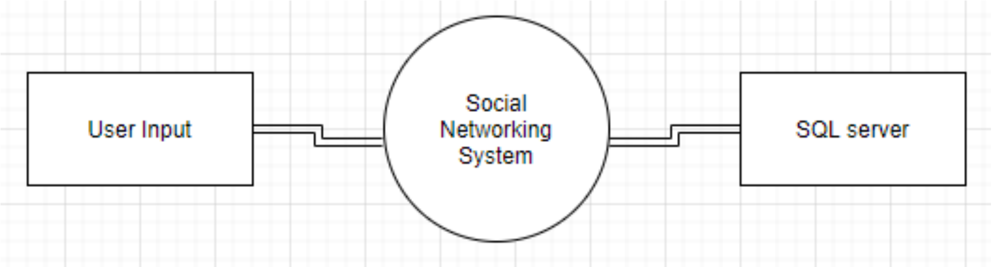
The drivers that will be focused on in this iteration are:

- QA-1: Performance
- QA-2: Availability
- QA-5: Usability
- QA-6: Security
- QA-7: Security
- CON-1: A user isn't able to create a collaboration project with only themselves.
- CON-3: Any changes to projects, profiles, or any other user change on the machine must be recorded.
- CON-4: The server must be accessed using one of the following web browsers (Chrome V3.0+, Firefox V4+, IE8+) and the only acceptable platforms being Windows,OSX, and Linux.

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

In this iteration, we will be refining the server side of the application. It's important for the system to support connecting to both client and the server easily. The modules which are required are located in the server.

The following components will be refined in this step:



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

Design Design and Locations	Rationale
Logically structure the client part of the system using the	**Web application** reference architecture supports deploying and running our application on a web browser. It is fairly easy to create views for a web application as compared to any other type of

<table>
<tr><td><b>Web application</b> reference architecture</td><td>application. Choosing this architecture itself supports QA-3, QA-4, QA-5. Moreover, it also supports CON-3 and CON-7.

<b><u>Discarded Alternatives:</u></b>

<table><tr><th>Alternative</th><th>Reason for discarding</th></tr><tr><td>Mobile Application</td><td>This alternative was discarded because this type of device was not considered for accessing the system.</td></tr></table></td></tr>
<tr><td>Logically structure the server part of the system similar to a web application</td><td>The database which is created for the purpose of this application is primarily used for storing different types of data that would be essential to our application. Most of the processing is done before-hand, meaning the data which is supplied by the user is first checked to make sure whether it's relevant or not, and once it's established that the data is relevant, it's stored in the database. The application follows a client-side scripting pattern.
All of the team members are familiar with the MySQL database system and hence we decided to proceed with this architecture specifically.</td></tr>
<tr><td>Physically structure the application using the three-tier deployment pattern</td><td>The three tier development pattern is used because the web application requires a browser that the user connects from for the client, a server that is used for the logic, and finally a dedicated server only for the data that is inputted on the application.

Discarded Alternatives:
A two tier development pattern was considered but was found to be insufficient due to the data having to remain on its own server. Furthermore, a tier development pattern over 3 would be too many as it's not required.</td></tr>
<tr><td>Build the user interface of the client application using HTML, CSS, bootstrap, JQuery, JavaScript and PHP</td><td>Building the user interface on the foundational blocks of HTML, CSS, bootstrap, Jquery, Javascript, and PHP provides a great outlook for the project as each one provides some aspect of workability upon the user interface.

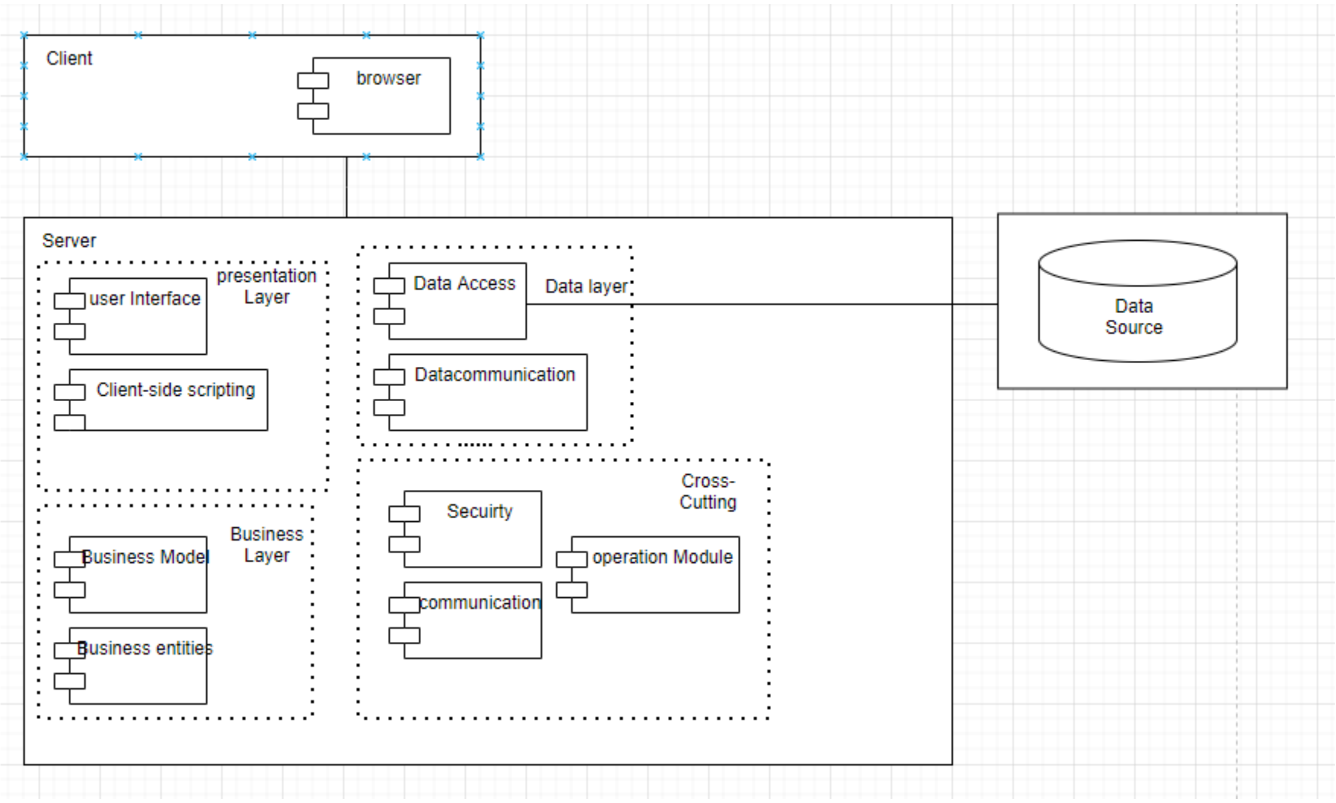
Discarded Alternatives:
React was discarded due to its complexity and learning curve. Angular was also discarded for the same reason of its complexity and the learning curve that comes along with it. The team of development felt more comfortable working with the frameworks listed above.</td></tr>
<tr><td>Deploy the application using GitHub Pages</td><td>GitHub Pages allows you to turn GitHub repositories to create websites.
All the changes are made and pushed in real time, which will allow</td></tr>
</table>

	changes to projects, profiles, or any other user change on the machine to be recorded. (CON-3) GitHub pages support more than 750 MME types hence, the deployed website will be accessible on many different browsers including Chrome V3.0+, Firefox V4+, IE8+. (CON-4)

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

Design decisions and Location	Rationale
Create initial domain	Initial domain model helps understand working of the system
Define modules based on Use case model	Business logic made using the reference from the use case model
Incorporate a separate layer to hold the data on server side	Adding a third layer to the server side that will manage the data ensuring that it is consistent with the data on the dedicated server.
Connecting the database access module to data source	The data layer that exists on the server side will have access to the database through an access module. The purpose of this module is to provide access to the SQL database server.

Step 6: Include all the Rough Diagrams



Step 7: Design the Kanban Board

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During 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-6		Selected reference architecture establishes the modules that will support this functionality.
QA-1			No relevant decision made, as it is necessary to identify the elements that participate in the use case that is associated with the scenario.
		QA-2	The use of GitHub Pages as a hosting platform will allow for minimal down time.
		QA-5	The use of a browser as a client in the web application architecture is sufficient to fulfill this quality attribute
QA-6			No relevant decisions have been made regarding the elements that participate in this quality attribute.
QA-7			No relevant decisions have been made regarding the elements that participate in this quality attribute.
		CON-1	Allowing users to collaborate on projects with others is sufficient to fulfill this concern
CON-2			No relevant decisions have been made regarding the elements that participate in this constraint.
		CON-3	Changes to projects, profiles, or any other user change on the machine is recorded
		CON-4	The server can be accessed using specific web browsers and the acceptable platforms are Windows, Linux and OSX.
	CON-5		No relevant decisions have been made regarding the elements that participate in this constraint.
		CRN-1	The final system is successfully completed in the allotted time
		CRN-2	The workload is evenly distributed within the project across the development team to fully utilize the team's capabilities.
		CRN-3	The first prototype of the system was successfully completed.