Delivery 2

1. Task 1: Identifying and finding inconsistencies in vision document

a. Defects:

Time spent during inspection: 3 hours

Defect #	Location	Defect type	Classification	Description	Status	Date corrected
1	Problem Statement (Section 2-2.1, Page 3)	Inadequacy	Major	Missing the '**affordable price**' aspect in the problem statement. It could have been correctly stated as "Renting tools and equipment that suit the customer's budget".		
2	Stakeholder Summary (Section 3-3.1, Page 4)	Omission	Major	One of the main stakeholders "Competitors" is omitted in the stakeholder summary.		
3	Assumptions and Dependencies (Section 4-4.2, Page 6)	Noise	Major	"Customers hold correct login credentials" can not be an assumption. Instead it is a requirement.		
4	Product Features (Section 5-5.3, Page 6)	Unfeasibility	Major	**History Tracking** - • This feature might be an overhead as it might hamper the overall project budget and timeline.		
5	Product Features	Noise	Minor	**Real-time data** - • This need not be the		

	(Section 5-5.5, Page 7)			website's feature. Instead, this would be the responsibility of branch employees to keep the website updated with the tools and equipment they have.		
6	Product Features (Section 5-5.21, Page 7)	Unfeasibility	Major	**View Recently Viewed Items** - • This feature would not only require the website to track users' navigation but also need to store & access this frequently. • This feature would take considerable amount of resources in terms of time and money and might not be feasible to be delivered in given timeline and/or budget.		
7	Product Features (Section 5-5.22, Page 7)	Unfeasibility	Major	**Report Generation** - • This feature would require an additional efforts for designing the database for keeping a track of rentals per customer and reporting on it. It might involve using third party technologies for report generation. • Therefore, this feature might require additional time and money to be invested on its implementation.		
8	Other Product Requirements (Section 6, Page 8)	Unmeasurability	Minor	• **Scalability requirement** describes that the website should handle a "decently high" load of user requests. • This should be quantified well for the technical team		

				to understand the expected requirement. (For example, 50,000 user requests simultaneously etc.)		
9	Other Product Requirements (Section 6, Page 8)	Ambiguity	Minor	• Instead of mentioning the **Performance requirement** qualitatively, it should be mentioned quantitatively. • For example, it could be alternatively stated as - "Average wait time for a server request should not exceed 3 seconds".		

b. Inconsistencies:

Time spent during inspection: 2 hours

#	Location	Inconsistency type	Classification	Description	Status	Date corrected
1	Problem Statement (***S1*** & ***S2*** - Section 2-2.1, Page 3)	Terminology Clash	Strong	• Instead of mentioning tools and equipment every time throughout the document, it would be better if it is mentioned in the glossary as "items".		
2	User Environment (***S3*** - Section 3-3.3, Page 5) and Assumptions and Dependencies (***S4*** - Section 4-4.2, Page	Structural Clash	Weak	• The phrase "stable internet connection" is mentioned at multiple places such as User Environment and Assumptions and Dependencies etc. However, the context in which it is used is the same.		

	6)					
3	User Environment (***S5*** - Section 3-3.3 & ***S6*** - Section 3-3.4, Page 5)	Terminology Clash	Weak	• It is mentioned that users should keep their browser updated for backward compatibility. • However, it should rather be for forward compatibility not backward compatibility.		
4	Product Features (***S7*** - Section 5-5.3, ***S8*** - Section 5-5.22, Page 6, 7)	Designation Clash	Strong	• Feature 5.3 defines that customers would be able to view their past rentals and feature 5.22 mentions that the backend data that is not creatable or viewable by customers. • These two statements clash with each other. If the software is designed to satisfy one feature then it would not satisfy the other. So, there seems to be a designation clash.		

2. Task 2: Documenting Conflicts

Time spent during inspection: 4 hours

- S1 - To create an online platform that provides a facility to rent ***tools and equipment***.
- S2 - These ***items*** are available at suitable prices to customers wherever and whenever they need it.
- S3 - Users are expected to have a working and stable internet connection and a web browser for using the website.
- S4 - User has a stable internet connection.
- S5 - Users should keep the web browser on their device updated to avoid any glitches pertaining to ***backward compatibility***.
- S6 - Website will be designed in a way taking into account possible ***future updates***.
- S7 - Users ***shall be able to see their past rentals*** along with the respective date and timestamp.

- S8 - ETR website shall allow only the system administrators to generate reports for management purposes based on rentals data available in the system's backend ***not creatable or viewable by customer and branch employee***.
- S9 - Modular design pattern shall be followed.
- S10 - As maintainability is directly proportional to the system design, the design should be flexible enough to accommodate future maintenance activities.

  Note: Statements S9 and S10 do not represent inconsistencies. They just overlap with each other since they talk about the same concept. Thus, they are not mentioned in Inconsistency Form.

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**Total**
S1	0	1000	1000	1000	1000	0	1000	0	0	0	5000
S2	1000	0	0	0	0	0	0	0	0	0	1000
S3	1000	0	0	1000	1000	0	0	0	0	0	3000
S4	1000	0	1000	0	1000	0	0	0	0	0	3000
S5	1000	0	1000	1000	0	1	0	0	0	0	3001
S6	0	0	0	0	1	0	0	0	1000	1000	2001
S7	1000	0	0	0	0	0	0	1	0	0	1001
S8	0	0	0	0	0	0	1	0	0	0	1
S9	0	0	0	0	0	1000	0	0	0	1000	1000
S10	0	0	0	0	0	1000	0	0	1000	0	1000
Total	5000	1000	3000	3000	3001	2001	1001	1	1000	1000	**20004**

In order to find the number of overlapping statements and conflicting statements, we divide the Total value from the above interaction matrix by 1000. The quotient gives the number of overlapping statements and the remainder represents the number of conflicting statements. Therefore,

Total / 1000 = 20004 / 1000 = 20.004
Number of overlapping statements = 20
Number of conflicting statements = 4

3. Task 3: Conflict Resolution

Time spent during inspection: 3.5 hours

a. Considering conflict between statements **S5** and **S6**. -

- S5 - Users should keep the web browser on their device updated to avoid any glitches pertaining to ***backward compatibility***.
- S6 - Website will be designed in a way taking into account possible ***future updates***.

1	Restore conflicting statements	• Here, the source of conflict is the browser not being updated. • So, the statement S5 could be stated as follows - *Future updates in the website shall be done taking into account backward compatibility with the earlier versions of browsers.*
2	Avoid Boundary Condition	• The boundary condition in this case is the outdated version of the browser. • This can be avoided by having a check in place to verify the current web browser's version when the website is loaded. If the version is not compatible with the required version then a relevant notification message could be displayed on the screen instead of loading the website.

b. Considering conflict between statements **S7** and **S8**. -

- S7 - Users ***shall be able to see their past rentals*** along with the respective date and timestamp.
- S8 - ETR website shall allow only the system administrators to generate reports for management purposes based on rentals data available in the system's backend ***not creatable or viewable by customer and branch employee***.

1	Specialize the conflict source or target	• Here, the users to which the data would be available is the source of conflict. • So, it could be specialized by clearly stating which users have access to what data. This would remove the ambiguity and therefore the conflict. These could be re-stated as follows. - **S7**: *Only **system administrators shall be able to see customer's past rentals** along with the respective date and timestamp.* **S8**: *ETR website shall allow only the system administrators to generate reports for management purposes based on rental **data that is only***

		accessible to System Administrators.
2	Introducing new requirement	• In addition to above method of resolution, this conflict can also be resolved by adding another requirement of User Authentication by means of which user trying to access past rental data shall be asked to enter a One-Time Password that would only be sent to the System Administrator by email. Thus, in this way anyone trying to access backend data would need to contact system admin in order to have the access. • However, this requirement needs to be assessed further considering the project budget and the timeline.

4. Task 4: Conflict Evaluation

Time spent during inspection: 1 hours

a. Analysis of alternate options for resolving conflict between **S5** and **S6**.

Evaluation Criteria NFR	**Significance weighting**	**Restore the conflict (Backward Compatibility)**	**Avoid boundary condition (Version Threshold Check)**
User Experience	0.4	0.9	0.7
Maintainability	0.3	0.6	0.8
Less Cost	0.3	0.4	0.8
Total	1	0.66	**0.76**

Conclusion - Avoiding Boundary Condition, that is, having a browser version check seems to be the optimal solution in this case.

b. Analysis of alternate options for resolving conflict between **S7** and **S8**.

Evaluation Criteria NFR	**Significance weighting**	**Specializing the source (Past data available only to System Admin)**	**Introducing new requirement**
Confidentiality	0.5	0.7	0.9
Fast Response	0.2	0.7	0.4

Less Cost	0.3	0.6	0.3
Total	1	**0.67**	0.62

Conclusion - Here, specializing the source of conflict seems as the better option than having a new requirement introduced in order to resolve the conflict. So, it is better to have the data access restricted to the system administrator rather than having it accessible to all users.

5. Task 5: Risk Management

Time spent during inspection: 6 hours

a. Risk Identification

i. Component Inspection

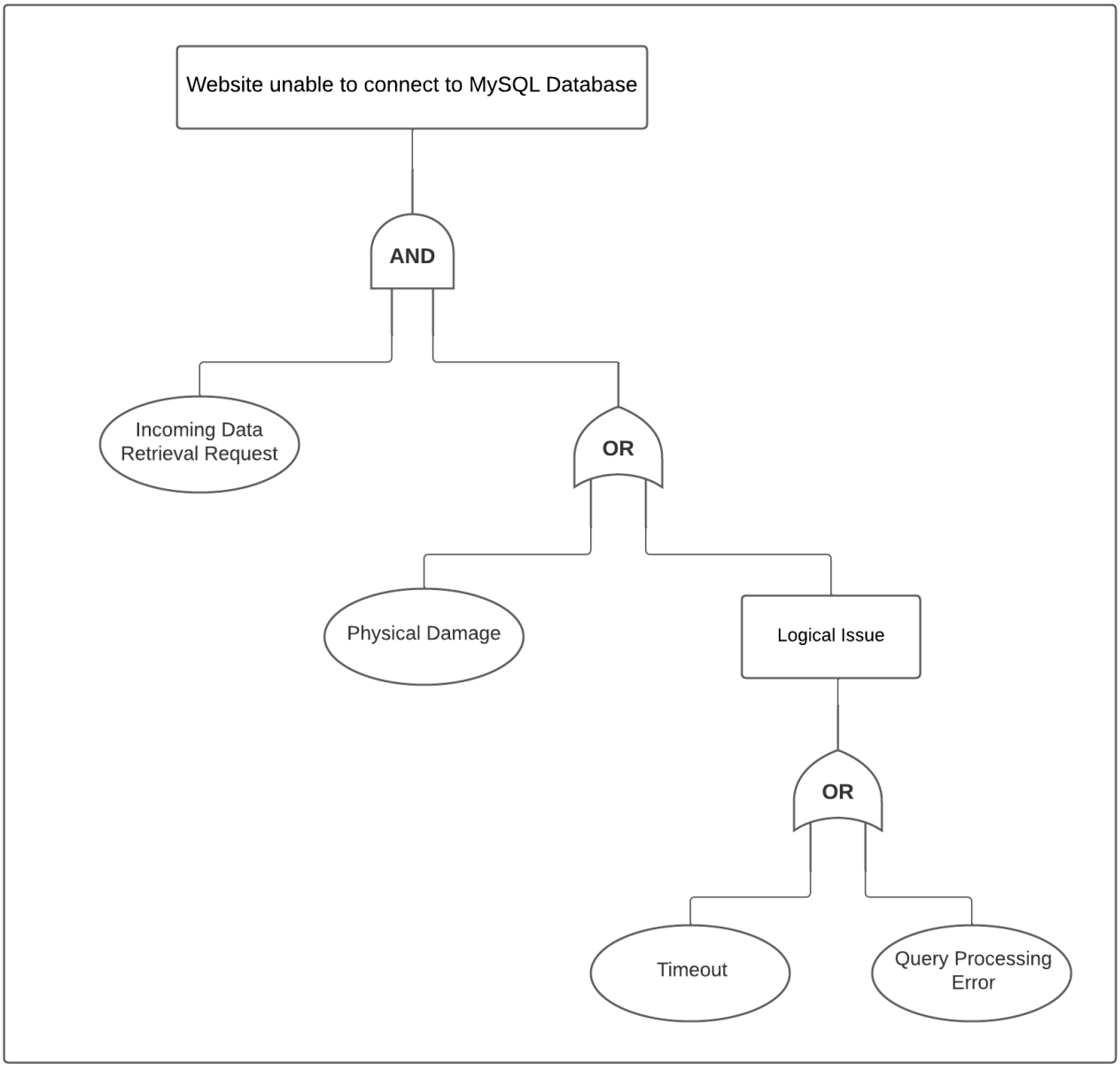
1. MySQL Server Connection Issue:
    - The server holding the database might fail in which case there will be a direct impact on the website's core functionality as data could not be retrieved in time.
    - Therefore, having multiple replicas of the database server could be an alternative option.

ii. Risk Checklist

1. Security Breach & Privacy Risk:
    - The website stores user's personal details and credentials. This information needs to be stored securely using cryptographic techniques to avoid any data breach.
    - In addition to this, the server on which this information is to be stored needs to be privately owned and should have an Authentication System in place to protect the sensitive information.
2. Inaccuracy:
    - If multiple users are trying to reserve the same tool or equipment at the same time then the website might not manage the conflict and might give false results to some or all users. So, the system has to accurately resolve this contention.
3. Downtime:
    - The website could be down in case of server failure or in the case when there are too many user requests to serve.
    - System should be protected against DDoS attacks to have the system available for the maximum possible amount of time.
4. Cost Overrun & Timeline Extension:

- The project could go beyond the budget if some of the low priority features are implemented. Therefore, feature priority needs to be decided. In addition, it might as well extend the due date of the deliverables.

iii. Risk Tree



b. Quantitative Assessment

i. For performing quantitative analysis of risks, an index called *Risk Exposure* is evaluated for every risk identified in the last step.

ii. Risk Exposure is essentially the product of likelihood of the risk and impact of that risk. Likelihood is basically the probability that the risk occurs and Impact is usually denoted in terms of monetary impact of the risk occurrence.

iii. Thus, the following table shows a quantitative analysis of all the identified risks by calculating Risk Exposure for each of them.

Risk	**Risk Likelihood (in %)**	**Risk Impact (in CAD)**	**Risk Exposure (RE_{before}) (Risk Impact * Risk Probability)**
MySQL Server Connection Issue	7	20000	1400
Security Breach & Privacy Risk	10	100000	10000
Inaccuracy	20	17500	3500
Downtime	20	60000	12000
Cost Overrun & Timeline Extension	25	30000	7500

c. Risk Control

A typical approach for risk control is to find countermeasures. There are several ways in which risk could be tackled with. They are -

a. Reduce risk likelihood
b. Avoid risk
c. Reduce consequence likelihood
d. Avoid risk consequence
e. Mitigate risk consequence

The risks in the above section are treated by identifying various countermeasures for each of the risks and then evaluating a ratio called *RRL* (Risk Reduction Leverage). It is basically the ratio of reduction in risk exposure to cost of the reduction. The countermeasure for which this ratio is greater than 1 is considered to be a cost-effective solution for reducing the risk.

$$Risk\ reduction\ leverage\ (RRL) = \frac{RE_{before} - RE_{after}}{Cost\ of\ risk\ reduction}$$

where,

RE_{before} = Risk Exposure before applying the countermeasure

RE_{after} = Risk Exposure after applying the countermeasure

Cost of risk reduction = Amount of money spent on implementing the countermeasure

Following are the risk control techniques for the identified risks. -

i. MySQL Server Connection Issue:

1. *Countermeasure 1 - Reduce Consequence Likelihood*: The likelihood of this risk could be reduced to a mere 2% by paying $950 for having a multi node database cluster in place instead of having a single node database. Even if it's not possible to connect to one node then other nodes could be queried for data.
2. *Countermeasure 2 - Mitigate Risk Consequence*: To pay roughly $500 on periodic data backups and thus having the risk likelihood reduced to 5%.

RE_{before} = 1400

	Likelihood (%)	Cost ($)	RE_{after}	$RE_{before} - RE_{after}$	RRL
Countermeasure 1	2	950	400	1000	**1.06**
Countermeasure 2	5	500	1000	400	0.8

Conclusion - From the RRL value, it is clear that the first countermeasure is more cost efficient than the second. So, implementing a multi-node database would be an optimal solution here.

ii. Security Breach & Privacy Risk

1. *Countermeasure 1 - Mitigate Risk Consequence*: Own a data server that is used solely for backing up the sensitive data. It could reduce the probability of authentication server failure by 4% at the cost of $4500.
2. *Countermeasure 2 - Avoid Risk*: Subscribe to third party authentication services and outsource this part of the functionality. It would cost around $3000 but would drop the risk likelihood to 7.5%.

RE_{before} = 10000

	Likelihood (%)	Cost ($)	RE_{after}	$RE_{before} - RE_{after}$	RRL
Countermeasure 1	4	4500	4000	6000	**1.34**

Countermeasure 2	7.5	3000	7500	2500	0.84

Conclusion - From the RRL value, it seems that having a dedicated data server for the purpose of backing up the data is the best option out of the two.

iii. Inaccuracy

1. *Countermeasure 1 - Avoid Risk*: One of the solutions is to implement a Wait List functionality wherein if multiple users request the same tool at the same time, only one customer gets to reserve the tool and others are queued in the Wait List. This decreases the contention likelihood to 15% with a cost of $800.
2. *Countermeasure 2 - Reduce Consequence Likelihood*: Another solution is to have a threshold on the number of available items. If the number falls below that threshold for a certain tool then disable online booking of that item and show a message to the user on the screen to contact the store for booking. Because this involves manual intervention at the last step, it is meant to reduce the likelihood to as less as 8% but having $2500 at the stake.

RE_{before} = 3500

	Likelihood (%)	**Cost ($)**	**RE_{after}**	**$RE_{before} - RE_{after}$**	**RRL**
Countermeasure 1	15	800	2625	875	**1.1**
Countermeasure 2	8	2500	1400	2100	0.84

Conclusion - Since RRL for first countermeasure is greater than 1 and RRL for countermeasure 2 is less than 1, it is clear that the former is more optimal than the latter.

iv. Downtime

1. *Countermeasure 1 - Reduce consequence likelihood*: Use various Network Monitoring Tools in order to identify normal and abnormal network traffic. This will keep the system updated with the incoming traffic ahead of time and will be able to alert the technical team to take appropriate actions if needed. This would require an investment of $2750 with a further reduced likelihood of 16%.

2. *Countermeasure 2 - Reduce risk likelihood*: Deploy Firewall in order to have customized strategies for various kinds of attacks. However, since this would require a specialized hardware and software infrastructure it would cost around $4200 but would be able to reduce the risk probability to 12%.

RE_{before} = 12000

	Likelihood (%)	Cost ($)	RE_{after}	$RE_{before} - RE_{after}$	RRL
Countermeasure 1	16	2750	9600	2400	0.88
Countermeasure 2	12	4200	7200	4800	**1.15**

Conclusion - From the RRL value, it seems that having a Firewall deployed for the website is the better option among the two.

v. Cost Overrun & Timeline Extension

1. *Countermeasure 1 - Reduce risk likelihood* : Features such as History Tracking, Report Generation could be skipped in the first version of the website thus focusing only on the main functionality of tool reservation and multiple browser support. There would be some efforts required from the stakeholders to prioritize the requirements. This would cost around $1600 but keeps the likelihood of risk occurrence to 20%.
2. *Countermeasure 2 - Avoid Risk*: The risk could be potentially avoided by having a team of experienced developers, tester, architect and project manager in the technical support team. By following an iterative development model, this risk could be reduced to 10% with an additional investment of approximately $3500.

RE_{before} = 7500

	Likelihood (%)	Cost ($)	RE_{after}	$RE_{before} - RE_{after}$	RRL
Countermeasure 1	20	1600	6000	1500	0.94
Countermeasure 2	10	3500	3000	4500	**1.29**

Conclusion - It is very clear from the RRL values of both the options that it is better to have expert technical resources develop the website reducing the overall risk probability.

6. Appendix

a. Time Log

Task	**Time Spent (in hours)**
Identifying Defects and Inconsistencies	5
Documenting Conflicts	4
Conflict Resolution	3.5
Conflict Evaluation	1
Risk Management	6