

Vision Document [ETR (electronic tool rental)]

1. Introduction

ETR is an application to rent tools and equipment for their improvement needs. This website allows us to rent carpet cleaner rentals, wood chipper rentals, lawn rollers, saws or any other tool for our project. This is actually a transformation from a conventional way of renting to renting it on a website. It's very convenient for people to rent things online rather than manually coming to rent. The objective of the application is crystal clear as we want to help people in the renting process by building an ETR(electronic tool rental) website.

ETR is mainly developed for individuals who want to rent tools and equipment. This application makes it easy as we have the tool you need at rental prices that suit the budget of the person. The website allows customers to check tool availability and pricing, and if it is adequate to make a reservation. With rental locations available in most of our stores nationwide, we make it easy to find just what you need, when you need it. This application will smoothen the renting process.

The development of the online system will focus on human interaction with computers by using interactive controls like buttons, links, images for easy navigation and information display. The website will have images of different tools, their prices, availability of the tool and the specified location. When we click on hyperlinks we will be navigated to different pages which makes it convenient to get the information of that particular tool. The output of the project would be the implementation of a real system that can be used by the people to rent their tools and equipment.

1.1 References

- [1]https://en.wikipedia.org/wiki/Vision_document#:~:text=A%20Vision%20Document%20is%20a.key%20needs%20and%20desired%20features.
- [2]<https://wikispaces.psu.edu/display/SASPMO/How%2Bto%2BDefine%2BProject%2BAssumptions%2C%2BConstraints%2C%2BDependencies%2Band%2BCritical%2BSuccess%2BFactors>
- [3]<https://app.diagrams.net/#>
- [4] <https://ca.shopwebly.com/>
- [5]<https://ca.gigapromo.com/>
- [6]<https://ca.searchley.com/>
- [7]<https://docs.google.com/document/d/17i4QFEry1qltFMGj0U51MPNoAiqOLnyTWxERHqWDtLw/edit>

2. Positioning

2.1. Problem Statement

The problem of	Renting tools and equipments online by providing the accurate information about pricing, availability
Affects	People who want to rent through this platform
The impact of which is	People would be able to rent things in a easy and efficient manner.

A successful solution would be	To create a website that makes life convenient for the people looking to rent by making the transition in the website clear for everybody to understand. In my perspective the UI/UX Designer is very important here.
--------------------------------	---

2.2. Product Position Statement

For	customer, system administrator, and branch employees
Who	People who want to rent the tools and equipment
The ETR(electronic tool rental)	is a software product
That	Helps customers to rent tools and equipments hassle free
Unlike	The conventional way of renting things
Our product	Is very convenient for the people to rent things as we have great website in place which helps customers to easily navigate to whatever they are looking to rent and check the availability and then subsequently rent it. The process of navigation through our product is easy.

3. Stakeholder Descriptions

3.1. Stakeholder Summary

Name	Description	Responsibilities
Product Manager	The project manager brings in the requirements to the team. Helps them understand the requirements and allocate the tasks to the team members for the development of the system. He even supervises the work done at easy point	1. Assigns and manages the tasks among the development team. 2. Interacts the people outside the team to get the queries sorted for the team and get the proper requirements. 3. Takes care of the time to deliver the project
Developer	He is the one who actually implements and maintains the system	1. Develop according to the requirements received. 2. Selects the best technologies to implement the system. 3. Implements the feature in chunks and then collaborate with other team members to complete the overall task.
Tester	The tester tests the deliverable system to check if the system is able to work in all test scenarios which prevents the system from crashing in production.	1. Tests each part of the system individually by creating and writing test cases. 2. Ensure that there are no bugs in the deliverable system. 3. Performs Regression and User Acceptance testing before release.

Investors	People who are funding this ETR Application.	1. Make sure that the budget of the project does not increase. 2. Fund management advice. 3. Creating confidentiality and goodwill.
Technical Architect	He specifies which programming language to consider, creates the design or flow for the entire system, defines the technical standard for code and platform	1. Work with the team to determine changes. 2. Estimate the project velocity. 3. Good program solving skills.

3.2. User Summary

Name	Description	Responsibilities	Stakeholder
Customers	They are the end users who actually use the system	Login for Register in the website for renting	Self represented
System Administrator	They are among the key stakeholders in the system.	They take care of adding new branches in the system and create users for the branch employees.	Self represented
Branch Employee	They are end users who use the system to take care of the inventory in respective locations	The branch employees are responsible for adding/removing tools for rental in their respective locations; they are also responsible to hand out the tools to the customers in store, and to receive them, for which they need to access the system to update the status of the tool (available, rented, out of order, not available).	Self represented

3.3. User Environment

- Customers, Administrators, branch employees are the end users of the system.
- Branch employees are the ones responsible for Adding, modifying the inventory of a particular location.
- Admin is the one who finally approves any kind of change. He is the one who publishes the inventory changes done by branch employees.
- All the tasks are performed sequentially as the branch employee will check the availability of tools in the location and add them to the systems. Admin confirms the changes of branch employees and publishes on the website and then the customers according to the need reserves the tool or equipment.
- The time spent may vary as it depends on the external factors such as network issues, blockers from stakeholders etc.

- Our application provides an easy way to rent things by providing an application which is easy to navigate. It's just a matter of a few clicks to reach a particular page and do an action correspondingly.
- It eases the work of customers and helps save a lot of time and energy.
- There are similar platforms like <https://ca.shopwebly.com/> <https://ca.gigapromo.com/> <https://ca.searchley.com/> etc.
- I guess it would be a good idea to integrate all such platforms to get the best out of everything.

3.4 Key Stakeholder or User Needs

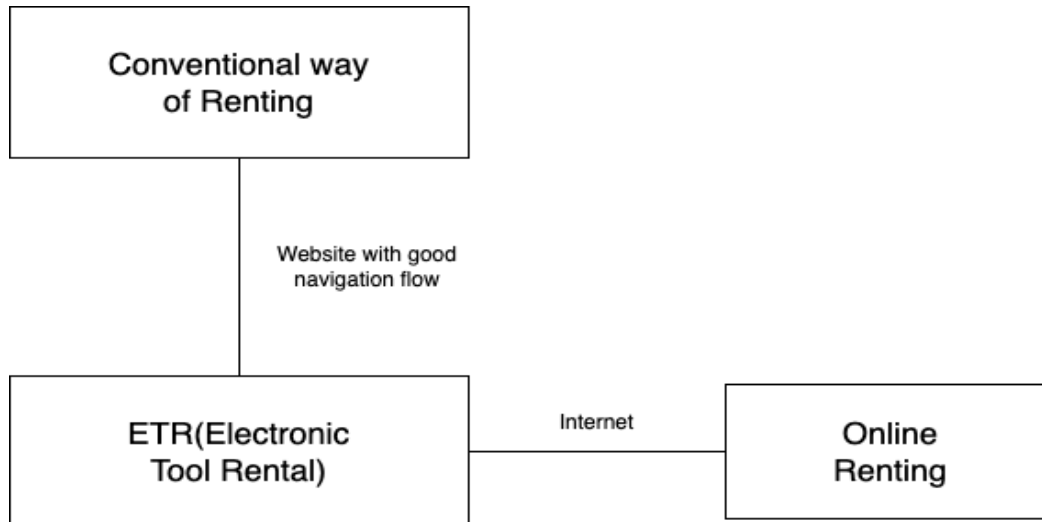
- ❖ Here we are developing a website for easy access of customers to rent tools and equipment.
- ❖ The inventory is properly adjusted by the branch employee to not cause any errors.
- ❖ Before publishing anything the administrator properly verifies the things.
- ❖ This web site is the best solution to the conventional way of renting things.

Need	Priority	Concerns	Current Solution	Proposed Solutions
Customer	High	Efficiency Interactive screens Easy of use	Conventional way of renting tools and equipment	Build a website for renting purpose
Receive updates of tools posted on website	High	Efficiency Response time Easy of use	None	Allow admin to get updates on adding the inventory by branch employees.
Receive notification on any modification needed on the content	High	Level of granularity	None	Allow branch employee to receive notification on modifications posted by Admin

4. Product Overview

4.1. Product Perspective

- ➔ ETR(Electronic Tool Rental) will change the way of renting as it's fast, easy and convenient.
- ➔ It is consistent as everyone has to go through the same process for renting or reserving tools.
- ➔ Customers just need a good internet connection to go through the website.



4.2. Assumptions and Dependencies

Assumptions	Dependencies
Customer has an active internet connection	We are dependent on hosting service for 24/7 availability
Client runs an Operating system which supports browsing	Website Interface must be convenient to use
Customers have their correct usernames and passwords to access the site	Information of all customers should be stored in a database to be accessible by the ETR website.
ETR website runs 24hrs/day	Remote/local servers should be running 24hrs/day

5. Product Features

1. **Log-In:** Customers, Branch employees and Admins should provide their username and password to access or reserve the tools. Based on the credentials different views of the website will be shown based on the permissions they have.
2. **Tools Information:** Customers will be able to access all the tools present in their location. They will be able to reserve whatever they like depending on the availability.
3. **Navigation menu for Branch Employees and Admins:** In our website we have different menus for the branch employees and admins where they can navigate to things like “inventory”, “Pending”, “Drafts”, “Published” and “Notifications”.
4. **Modifications:** Once Admin reviews the content posted by the employee. He will post a feedback or a comment if anything is wrong. The employee then can modify the things according to the admin instructions. This way we have double surety of data.
5. **Comment summary:** Admin can see the content posted by the employee and if everything is perfect then we will post the content on the website.

6. Other Product Requirements

- ★ **Applicable standards:** This is an PHP MVC based project with MYSQL database structure. Here we are using HTML5, CSS3, Angular for frontend which is easy for access, PHP as a middleware to connect to backend MySQL server. We are using Cloudinary for storing Content.

- ★ **Hardware requirements:** Computer with proper operating system such as windows, LINUX or MAC and internet connection.
- ★ **Performance requirements:** Server should be capable of handling more than 200K end-users simultaneously. System should send responses immediately on click of every operation. As we are using Cloud if any faults occur in future then the server will handle it without reducing the performance.
- ★ **Environmental requirements:** VSCode as an IDE with chrome browser.
- ★ **Documentation requirements:**
 - o **User Manual:** Not required as the application is easy to use since we are using advanced framework like Angular
 - o **Online help:** Context-specific and general help will be available for all functions within the system.
 - o **Installation:** Configuration Guide, Read Me File. An installation manual will be provided.
 - o **Package:** HTML & CSS support with Angular 2.0+ version. PHP and MySQL.

Priority of mentioned requirements: HTML,CSS along with Angular will make application frontend as standalone/Stable. Moreover, to reduce the risk of vulnerabilities we use PHP to transfer the data from the frontend and process it towards the server/backend. It also improves security as the frontend and backend tier will not have direct connection.

Task 0: Logging:

It took me 15hrs to write the Vision Document. The times with respect to the headings is as follows:

1. Introduction: 2hrs
2. Positioning: 5hrs
3. Stakeholder Description: 3hrs
4. Product overview: 2hrs
5. Product features: 2hr
6. Other product requirements: 1hr

Task 0: Logging

I have used google docs for logging hours but i haven't included the link here due to anonymity issues.

S.No	Tasks	Time Taken
1.	Identifying and finding inconsistencies in the vision document	10hrs
2.	Documenting Conflicts	4hrs
3.	Conflict Resolution	5hrs
4.	Conflict Evaluation	5hrs
5.	Risk Management	10hrs

Task 1: Identifying and finding inconsistencies in vision document

1. Defects Table

Defect #	Location	Defect type	Classification	Description	Status	Date corrected
1	Introduction	Major	Omission	Problem world feature is not stated anywhere in the introduction or scope	Closed	29-07-2022
2	Introduction	Major	Inadequacy	Requirement item in the introduction is not adequately stating a problem world feature.	Closed	29-07-2022
3	Introduction	Major	Overspecification	Introduction is having information which is not appropriate to the problem statement mentioned.	Closed	29-07-2022

4	Positioning – Problem Statement	Minor	Poor Structuring	Statements provided are not structured properly which does not make sense to the user	Closed	29-07-2022
5	Stakeholder Description – User Summary	Major	Opacity	Description not yielding any information about the product.	Closed	29-07-2022
6	Stakeholder Description- User Summary	Major	Ambiguity	Employee and Admin responsibilities are ambiguous where one feature interprets in multiple ways.	Closed	29-07-2022
7	Positioning- Problem Statement, Introduction, Stakeholder Description -User Environment and Other Product requirements	Major	Unmeasurability	In the document it is not precisely specified that the document is efficient.	Closed	29-07-2022
8	Stakeholders -User Summary	Major	Contradiction	Two stakeholder(Employee and Admin) responsibilities are mentioned in an incompatible way.	Closed	29-07-2022
9	Introduction, Product Features	Minor	Unintelligibility	Document provided is incoherent/Unclear to the user to read about functionalities of end-users and purpose of the etr(electronic rental tool) application.	Closed	29-07-2022
10	Other Product requirements	Major	Opacity	The dependencies in the technologies is not mentioned clearly.	Closed	29-07-2022

11	Stakeholder Descriptions- User Summary, User Environment , Product Features	Major	Poor Modifiability	Documented requirements and responsibilities need to be modified globally throughout the document.	Closed	29-07-2022
12	Other Product Requirements	Major	Unfeasibility	Technologies used to develop applications can exceed the budget or the system will be deployed beyond the scheduled dates.	Closed	29-07-2022

2. Inconsistency table

#	Location	Inconsistency type	Classification	Description	Status	Date corrected
S1	Stakeholders- User Summary	Structure Clash with S2	Strong	Employees add the available items in the inventory (Designation Clash with S2)	Closed	29-07-2022
S2	Stakeholders-User Summary	Structure Clash with S1 and S3	Strong	Administrators add and approve the items in the inventory (Designation Clash with S1 and Structure Clash with S3)	Closed	29-07-2022
S3	Product Features- Updates	Structure Clash with S2 and Terminology Clash with S4	Strong	Employee can modify the inventory and can publish it	Closed	29-07-2022
S4	Stakeholders -User Summary	Terminology Clash with S3	Strong	Admin can modify the content released by the employee, approve, and then publish the inventory changes updated by the employee.	Closed	29-07-2022

S5	Stakeholders -User Summary	Designation Clash with S2	Strong	Customers rent the tools according to the availability in their location.	Closed	29-07-2022
----	----------------------------	---------------------------	--------	---	--------	------------

Task 2: Documenting Conflicts

We try to build the interaction matrix (Kotonya & Sommerville, 1997) based on the information generated in task 1. Below are the statement that are elicited from ETR(Electronic Rental Tool) Vision document:

S1: Employees add the inventory of the available tools in a particular location

S2: Administrators publish the inventory details so that the customers can rent it.

S3: Employee modifies inventory accordingly to the location and publishes it.

S4: Admin can modify the inventory released by Branch Employee, approves, and then publishes the changes updated by the employee.

S5: Customers can rent the tools according to the availability in their location.

Statements	S1	S2	S3	S4	S5	Total
S1	0	1	1000	0	1000	2001
S2	1	0	1	1000	1	1003
S3	1000	1	0	1	0	1002
S4	0	1000	1	0	0	1001
S5	1000	1	0	0	0	1001
Total	2001	1003	1002	1001	1001	6008

The total number of non-conflicting statements and conflicts is given by quotient and remainder of integer division of Total | Total by 1000

In our matrix it is, $6008/1000 = 6.008$. So, The Number of conflict = 8 and The Number of non-conflicting overlap = 6

Task 3: Conflict resolution

1. S1 with S2 : Here Employee as well as Admin is publishing the Inventory based on availability in a location.

To reduce this conflict, “**Specialize conflict source or target**” resolution technique can be used where initially we have to identify the target, in our case Customers are the targeted end-users where we have to verify from which source Customers are able to receive the updates of availability of a particular tool from the portal. In our rental tool platform admin end-users have the authority to change inventory which further customer can be able to see and book accordingly. We can rewrite the statement as “**Employees update the inventory after which customers can view only after the Admin publishes/approves the inventory**”.

Another way is using “**Questionnaires**” which is one of the artifact-driven elicitation techniques. Using this technique, we can ask a specific list of questions for stakeholders that includes all kinds of end-users who have more system knowledge. Based on the response received we can evaluate and document the functionalities respectively.,

1. S2 with S3: Here in Statement S2 Admin is publishing inventory which is accessed by customer and in statement S3 it is written as Employee Publishes inventory

We can reduce this conflicting statements by using “**Restore Conflicting Statements**” resolution technique through which we can combine both statements together and rewrite the statement as “**Employee updates the inventory if Admin raises any issues related to inventory, employee modifies and updates inventory which then will be published by Admin**”.

Another way is using elicitation technique “**Card Sorts**” where we can sort all the product features separately according to the end-users. For example: We have to first document all the functionalities in ETR(Electronic Tool Rental) Portal such as login, Inventory information, Rental Track, Published, Pending, Drafts etc.,after documenting we have to sort functionalities according to end-users (Customer, Admin and Employee).

2. S2 with S5: Here in S5 Customers are renting the tools that are updated by the employee but in S2 it is mentioned as Admin is publishing inventory and slides for customers.

To reduce this conflict we can use the “**Drop lower-priority statements**” operator through which we can reduce the priority of conflicting statements. For Example: we can reduce S5 priority by removing words that are conflicting with S2 and rewriting sentence as “**Customers rent the tools once they have been approved and published by Admin**” After rewriting the sentence as mentioned we can say both the statements S2 and S5 are non-conflicting.

Another way is using "**Weaken Conflicting statements**" Operator where we have to lower the restrictions on one or both conflicting statements. In our conflicting statement we can modify S2 as "**Admin Publishes the inventory for customers which are updated by employees**". This way we can reduce conflict between S5 and S2 where in S2 if we mention that the employee is the one who updates the inventory and the customers can access the tools properly.

3. S3 with S4: In S3 we have written as employee modifies the inventory and publishes it on other hand S4 talks about the same functionality but end-user as Admin.

By using "**Restore Conflicting statements**" Operator we can reduce the conflict among the statements mentioned. Best way is to combine both the statements and rewrite a single statement as "**Employee modifies the content which admin comments and after modification is done by employee Admin gets notification and (S)he publishes the inventory which is later accessed by customers**".

We can also reduce conflict using "**Specialize conflict source or target**" technique where initially we have to identify source or target users in the system. In our system we are talking about modifying information by both end-users Admin and Employee, in which we consider these end users as a source. Instead of just modifying in S3 we can modify the statement as "**Employee modifies content which admin asks to modify and further publishes**" so that S3 is transformed into a conflict free version.

Task 4: Conflict evaluation

Here we are calculating the best suitable resolution techniques or operators based on the degree of contribution to the various non-functional requirements that are elicited or, their degree of resolution of identified conflicts. We use Weighted matrices for quantitative reasoning techniques. For evaluating the matrices will use below formulae:

$$\text{totalScore(opt)} = (\text{Scores(opt,crit)} \times \text{Weight(crit)})$$

Evaluation Criteria NFR	Significance Weightage	Conflict resolution by specializing source or target	Conflict resolution by sorting	Conflict resolution by restoration
Accuracy rate	0.1	0.5	0.5	0.9
Fast response	0.2	0.3	0.7	0.6
Data Integrity	0.3	0.6	0.3	0.7
Reliability	0.4	0.7	0.4	0.4

Total	1.0	0.57	0.44	0.58
--------------	-----	------	------	------

1. Calculating weightage for resolution by specializing source or target is

$$\text{totalScore(opt)} = ((0.5*0.1)+(0.3*0.2)+(0.6*0.3)+(0.7*0.4)) \\ = 0.57$$

2. Calculating weightage for resolution by sorting

$$\text{totalScore(opt)} = ((0.5*0.1)+(0.7*0.2)+(0.3*0.3)+(0.4*0.4)) \\ = 0.44$$

3. Calculating Weightage for resolution by restoration

$$\text{totalScore(opt)} = ((0.9*0.1)+(0.6*0.2)+(0.7*0.3)+(0.4*0.4)) \\ = 0.58$$

Conflict resolution by restoration and by specializing source or target are similarly resolving the conflicts present in the document. On the other hand resolving conflicts using sorting may not perfectly resolve issues in 50% of cases.

Task 5: Risk Management

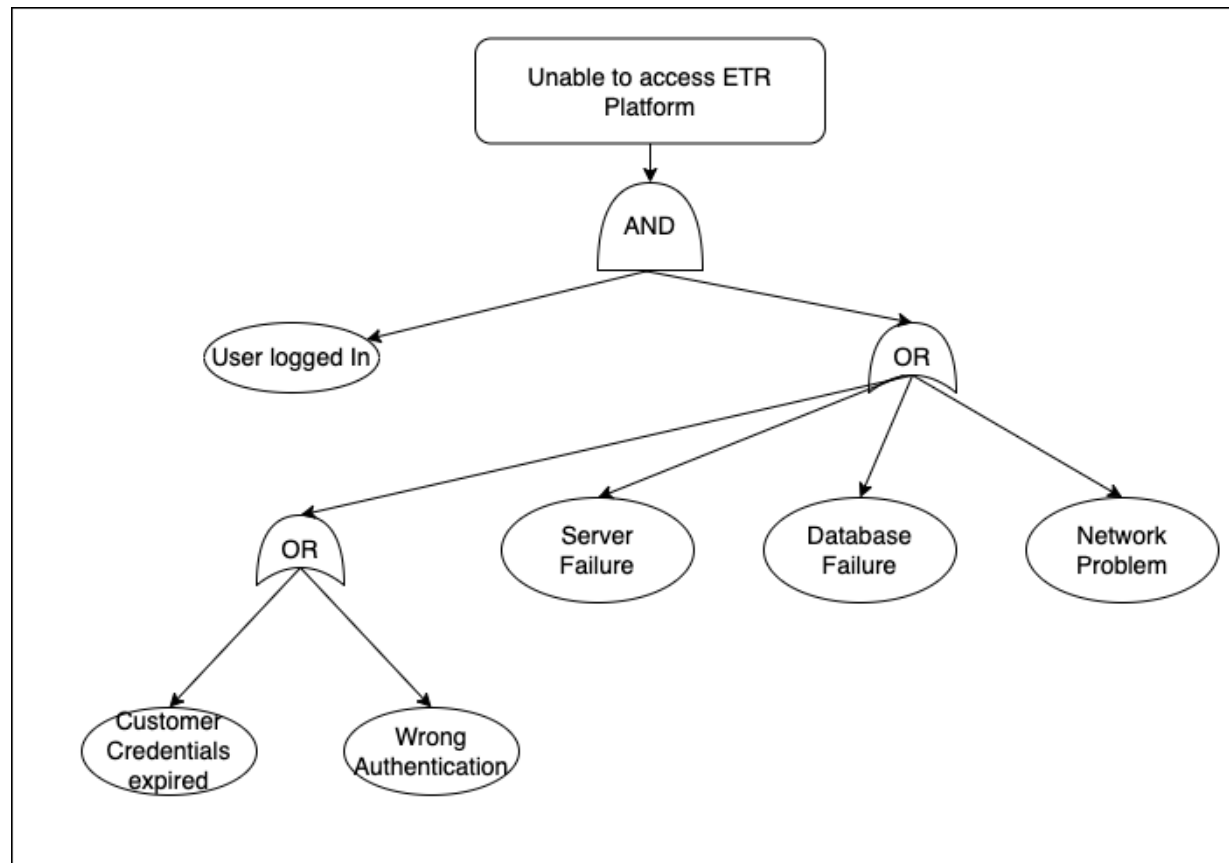
1. Risk Identification

a. Component Inspection

- i. SQL Server: This is the place where we actually store the data related to tools for our platform. so, the risks we can identify in the backend server are performance issues, Data Security, Data Loss and SQL Injection.
- ii. Frontend system (PHP): We have used PHP MVC as our frontend which is then embedded with HTML and CSS. Below are the risks identified in frontend are Source code revelation, Remote file inclusion and session hijacking.
- iii. Hardware Risks: Our application will run on a Computer with a proper operating system such as windows, LINUX or MAC and internet connection. The risks can be identified as Malfunction of electro-mechanical components and Power fluctuations and failures.

- b. Risk Checklists: Risk Checklists are the one which can be built from risk categories that negatively impact the requirements of the system. Checklists include various elicitation criteria that depends on non-functional requirements of the system such as Performance, deadlock conditions, integrity etc.,

- c. Risk Trees: The identification of risks through component inspection can be made more systematic by the use of risk trees. Below is the risk tree constructed for faults that occur in our system:



2. Risk Assessment

RISK: "ETR(Electronic Tool Rental) server does not respond."

Consequences	Likely	Possible	Unlikely
Forbidden access	10	9	7
Voluminous Data	10	9	7
Institution Reputation Damaged	5	3	3
Number of users decreased	8	7	3
Increased Response time	8	7	5

3. Risk Control

Using the risk reduction tactics above mentioned, the risk “ETR server does not respond” can be reduced in various ways:

- a) Reduce risk likelihood: Have frequent backup servers that gives the response in case the main server goes down. Along with the backup servers there can be introduced replicas of the database in a parallel server. Replica will authenticate the user and load the data based on the type of user in case the main server goes down.
- b) Avoid risk: To avoid the risk, troubleshoot the server and frequently check if it becomes unresponsive and demands immediate attention.
- c) Mitigate risk consequence: In case of the error 403 when authentication error occurs, the error should be presented in the user interactive way so that user understands what happened and recheck his/her credentials.
- d) Reduce consequence likelihood: To reduce the likelihood of the consequence Error 500 (error that occurs due to voluminous data of the user), the data of the user can be separated on different servers so that one server does not get overloaded and ends up giving no response.
- e) Avoid risk consequence: The increased response time is a consequence of the risk that impacts user experience and can be avoided by optimizing the database. Slow queries to retrieve the data from the database is one of the major reasons for the poor performance of the system and this can be avoided by rewriting queries, using indexes, and changing schemas.

There are two methods to reduce the risk likelihood:

- 1) Option one is to have backup servers.
- 2) Second is to have replicas of the server.

In the first option it is estimated that risk will be reduced by 15% and cost is \$4000. However, in second the risk will be reduced by 5% and cost incurred to reduce the risk will be \$5000. Let's assume that the risk of the server going down is 25% and the cost to fix it will be \$40000.

Based on the above options the risk reduction leverage in first will be

Risk Reduction Leverage (RRL) = $RE(\text{before}) - RE(\text{after}) / \text{cost of risk reduction}$

$$1) RRL1 = 40000(25-15/100) / 4000 = 1$$

$$2) RRL2 = 40000(25-5/100) / 5000 = 1.6$$

Based on the risk reduction leverage calculated above the option 1 would be more suitable which is having backup servers.

Response to reviewers, teacher assistant and instructor.

1. Introduction:

I would like to thank the reviewers and teacher assistant for their detailed feedback and useful suggestions to improve my vision document.

I have carefully considered all the issues raised by my peers. Teacher assistant, and instructor and prepared a revised vision document. This document outlines how I have addressed each comment individually. Each comment has been assigned a number R(1-3).C(1-N), where the number to the right of the R identifies the reviewer, and the number to the right of the C identifies the comment. Received 8 comments from R1 peer, 5 for R2 peer and 1 from TA. All the comments I have addressed in the following section 3.

My response to each comment is highlighted in blue. Thanks for the opportunity to improve my vision document.

Sincerely,
Priyanka Kudumula

2. Create a table when you identify each reviewer with a unique identifier.

Reviewer	ID
Peer 1 (8 comments received in D2)	R1
Peer 2 (5 comments received in D2)	R2
Teacher assistant (1 comment received in D1)	R3

3. Reviewer comments

R1.01 [The student didn't document the below defects: Location: 5. Product Features, Location: 2.1 Positioning - Problem Statement : "In my perspective the UI/UX Designer is very important here." etc]
[Agreed, I have added the defects in the vision document.](#)

R1.02 [Inconsistency table issues – In S1 inconsistency, the student has "Structure Clash with S2" as Inconsistency type, but in the description, he/she says "Designation Clash with S2". Therefore, is it a Structure clash or a Designation Clash or both? The clash should be documented in the Inconsistency Type only.]
[Agreed and changed to structure clash](#)

R1.03[The student is missing the below inconsistencies in his/her inconsistency table: – Location: 3.4 Key Stakeholder or User Needs: "the administrator properly verifies the things", "conventional way of renting things"]
[Agreed and added the inconsistency](#)

R1.04[**The student did not evaluate all the conflicts. Indeed, he/she did not do a weighted matrix for each conflict's alternative options.**]

I do not agree with this feedback given because, in document I have evaluated the weights properly.

R1.05[**Risk Identification** – b. Risk Checklist]

I don't agree as the Risk Checklist itself means that we need to make a checklist of the risks.

R1.06[**Risk Identification** - c. Risk Tree: Customer credentials expired and wrong authentication are not risks. Indeed, they should not be placed in the risk tree.]

I have removed that from the document

R1.07 [**Risk Assessment** - he/she provided a table stating the likelihood of each 'consequence', which is not what should be done in this section.]

Agreed and removed it.

R1.08 [**Risk Control**: The student did not propose at least 2 countermeasures for each risk. Instead, he/she proposed 2 countermeasures for 1 risk only and not for the remaining risks.]

Agreed and added.

R2.01 [**Defects 2**: Introduction is too lengthy. There is no adequacy rather it should be overspecification]

I don't agree to this as there were certain parts missing in the introduction which could have been clearly illustrated so i mentioned as Inadequacy.

R2.02[**Positioning**: the Problem of another inconsistency is opacity]

Agreed and changed

R2.03[**Conflict Resolution**: Conflict resolution tactics have not been specified properly.]

I have mentioned the conflict resolution tactics clearly now.

R2.04[**Conflict Resolution**: This section is incomplete and inadequate]

Changed this as per reviewer 1 suggestion already.

R2.05[**Risk Assessment**: Missing: Reasoning/description of each risk]

I have added the description now

R3.01 [Some missing stakeholders like competitors that are already available in market also some stakeholders. Key stakeholders missing key components such as Data Security,Record keeping.some missing product features such as password recovery]

Agreed and made the necessary changes in the document.