

Rently Risk Management Plan (RMP)

This management plan is designed to track, analyze and manage the identified risks associated with development of the Rently condominium management system.

Scope

This RMP document will outline the risks identified by the Rently development team, their likelihood, severity, and chosen response strategy.

Due to lack of time and allocated budget, this document will show the aforementioned information in the form of tables and matrices.

This document will not mention the specifics of how and who will manage the risks identified.

Disclaimer

All risks are assumed to be distributed evenly among the members of the respective subteam (frontend, backend). Program risks are taken on by the team in its entirety. Delegation of risk management is done at the development team's discretion.

Any questions regarding the risks may be asked at the stakeholder meeting directly preceding the addition or adjustment of the risks

1.0 Risk Identification

The first step in analyzing and managing risks associated with a project is to identify them correctly. Similar to requirements management, misidentification of risks can lead to, at best, ambiguity and a partial understanding of the risk and its chosen mitigation technique. Usually, risk management is best done by a subset of the team, typically those in positions of authority. This is because full understanding of the product being developed is essential to identifying the risks with the least amount of misunderstandings. It is also done by managers and leads so developers and technicians need not concern themselves with the project as a whole.

In the case of a typical student-led software development project, it is not uncommon for all students to have an understanding of the entirety of the system being developed. In this case, choosing 2 or more teammates to perform the risk analysis may be sufficient, since the students can discuss and include their differing points of view.

In the case of the Rently system, two students of the 9-person team identified the risks for following sprints.

2.0 Risk Analysis

Risk analysis on the Rently system was done with a clear understanding of the values being chosen.

The “Likelihood” and “Impact” values associated with each risk were done according to the image shown in Figure 1 below.

Risk Matrix & Scales		
LIKELIHOOD SCALE		
Level	Definition	Likelihood
1	It would be surprising if this happened	10%
2	Less likely to happen than not	30%
3	Just as likely to happen as not	50%
4	More likely to happen than not	70%
5	It would be surprising if this did not happen	90%
IMPACT SCALE		
Level	Definition	Cost (% of WBS element value)
1	<ul style="list-style-type: none"> Schedule: Insignificant or no schedule slippage. Functionality: “Functionality” decrease barely noticeable or no impact. Programmatic: Only secondary project objectives could be impacted. Quality: “Quality” degradation barely noticeable 	10%
2	<ul style="list-style-type: none"> Schedule: 5% slippage. Additional activities required. Able to meet need dates. Functionality: Minor areas of “Functionality” are affected. Same approach retained. Programmatic: Some main project objectives could not be met. Threat can most likely be eliminated with workarounds. Quality: Only very demanding applications are affected 	30%
3	<ul style="list-style-type: none"> Schedule: Overall project 5-10% slippage. Some milestone slips. Functionality: Moderate areas of “Functionality” are affected but workarounds available. Programmatic: Main project objectives could not be met. Threat can likely be eliminated with workarounds. Quality: “Quality” reduction requires client approval. 	50%
4	<ul style="list-style-type: none"> Schedule: Overall project 10-20% slippage. Possible project critical path impacted. Functionality: Major areas of “Functionality” are affected but workarounds available. Programmatic: Main project objectives most likely will not be met. Workarounds still available. Quality: “Quality” reduction most likely unacceptable to the client approval. 	70%
5	<ul style="list-style-type: none"> Schedule: Overall project >20% slippage. Very likely major project milestones can't be met. Functionality: “Functionality” reduction unacceptable to client. Project end item is effectively useless. Programmatic: Main project objectives can't be met without major changes. Quality: “Quality” reduction is unacceptable. Project end item is effectively unusable. 	90%

Figure 1: Values associated with risk likelihood and impact

3.0 Risk Matrix

This section is going to cover the uses and the development of the Risk Assessment Matrix that will be used and updated throughout the project development life cycle.

A risk matrix is typically used to give a quick view of the number of risks, how impactful they are and how likely each risk is. This can give a good overview of the system as a whole as well, for example if all risks were located in high-likelihood, high-impact cells, it could warn project managers and stakeholders about the project as a whole.

Figure 2 shows the risk matrix, along with the identified risks placed in their respective cells.

Risk Matrix						
LIKELIHOOD	5: 90%					
	4: 70%					
	3: 50%					
	2: 30%			1,2,3,11	8,9,10	
	1: 10%		6	4,5	7	
		1: Insignificant	2: Minor	3: Moderate	4: Major	5: Severe
		IMPACT				

Figure 2: Rently Risk Matrix with risk IDs

All risks are located below “Severe” in impact and all have a likelihood below 50% of occurring. This is mainly due to the close communication with the team, and the experience that some students have in web development.

The identified risks each have a unique identifier, this can help clear issues when the team discusses mitigation techniques, preventing ambiguities between risks that may be worded the same.

4.0 Risks & Mitigation Strategies

Typically, there are 4 risk mitigation strategies used involved in project management:

- Accept
- Avoid
- Mitigate
- Transfer

Accept: Acknowledge the risk and choose not to resolve, transfer or mitigate. Typically if the risk is of low impact or can simply not be managed

Avoid: Completely eliminate or forego risk, typically by not engaging in activities that can lead to the occurrence of the risk

Mitigate: Reduce the likelihood or impact of the risk

Transfer: Assign or move the risk to a third-party

These are the mitigation strategies that are going to be used in the risk analysis process.

5.0 Risk Analysis

Here we will cover the identified risks, their likelihood, impact and the associated mitigation strategy.

Table 1: Sprint 1 Risk Analysis

Risk ID	Description	Impact	Probability	Severity	Entry Date (Sprint)	Response Strategy	Response Plan
1	Management Lack of team communication	3	2	Low	2/3/2024 (1)	Mitigate	Communication by slack and discord. Multiple meetings per week if needed
2	Technical Features not implemented	3	2	Low	2/3/2024 (2,3,4)	Mitigate	Most important features are implemented first
3	Management Team members not completing tasks	3	2	Low	2/3/2024 (1,2,3,4,5)	Mitigate	Team members will ask for help if a task is too time consuming
4	Technical Team members don't know the technologies being used	3	1	Low	2/3/2024 (1,2,3)	Avoid	Team members have discussed the technologies they know before beginning sprint 1
5	Management Development becomes out-of-scope of stakeholder needs	3	1	Low	2/3/2024 (1,2,3,4,5)	Avoid	Project scope is set and is to be followed to avoid development runoff
6	Technical UI not consistent site-wide	2	1	Low	2/5/2024 (2,3,4,5)	Avoid	Regular code reviews, developers have access to entire codebase. Sanity checks. Shared React components

7	Management Not understanding stakeholder requirements	4	1	Low	2/6/2024 (2,3,4,5)	Avoid	Multiple sprints and meetings with stakeholders throughout project development
8	Technical UI not connected to backend logic	4	2	Medium	2/6/2024 (2,3,4)	Mitigate	Performing tests on integrated system
9	Technical Backend logic not connected to database	4	2	Medium	2/6/2024 (2,3,4)	Mitigate	Performing tests on integrated system
10	Technical External Site hosting issues	4	2	Medium	2/6/2024 (4)	Mitigate	Transfer as well Performing tests on hosted system to ensure site reliability
11	Technical External Integration with 3rd party systems	3	2	Low	2/6/2024 (2,3,4)	Mitigate	Transfer as well Site hosting, 3rd party APIs, database management. Performing tests and checks throughout development