SE Experiment-9(Batch-C/D)

-Team Details-
Saina Hamid – UID (2024301008)
Kalpak Patil– UID (2024301020)
T.E. Computer Engineering – C

Aim:

Perform Risk analysis:

1)Create Risk table and calculate Risk Exposure for every risk.

(Risk ID, Description, Category, Probability, Impact in terms of Rs, Risk Exposure, Risk Response)

2)Create RMMM plan (risk information sheet) for 2 project specific risks.

Theory

What is Risk Analysis?

Risk Analysis is a formal project management process used to identify, assess, and plan for potential problems ("risks") before they happen. A risk is any uncertain event that, if it occurs, could have a positive or negative effect on the project's objectives, such as its schedule, budget, or quality. The goal of risk analysis is not to eliminate all risks, but to minimize their impact and maximize opportunities.

Who Uses It?

- **Project Managers:** They are primarily responsible for creating and maintaining the risk plan.
- **The Project Team:** Developers, designers, and testers are essential for identifying technical risks based on their expertise.
- **Stakeholders:** Clients and end-users (like the students and Admin for Collab Nest) help identify business risks, such as low adoption or usability problems.

Importance of Risk Analysis

Its primary importance is to move a project from being *reactive* (fixing problems after they occur) to being *proactive* (preventing problems before they start). A good risk analysis:

- Prevents project failure by identifying major threats early.
- Saves time and money by having a contingency plan ready.
- Improves communication and transparency by making everyone aware of the challenges.
- Increases the probability of project success.

Key Components of Risk Analysis

1. **Risk Identification:** Brainstorming all potential risks (e.g., "What could go wrong?").
2. **Risk Assessment:** Analyzing each risk's **Probability** (how likely is it to happen?) and **Impact** (how bad will it be if it does?). This is used to calculate the **Risk Exposure (RE)** and prioritize which risks to focus on.
3. **Risk Response Planning:** Deciding on a strategy for each major risk:
   - **Mitigate:** Take steps to reduce the probability or impact (e.g., do more user testing to *mitigate* low adoption).
   - **Avoid:** Change the project plan to eliminate the risk (e.g., remove a complex feature).
   - **Accept:** Acknowledge the risk and do nothing (used for low-priority risks).
   - **Transfer:** Shift the risk to a third party (e.g., using a reliable cloud provider like Firebase).
4. **Risk Monitoring:** Actively watching for "triggers" that indicate a risk is becoming a reality and then executing the response plan (the RMMM plan).

2. Risk Table

Risk ID	Description	Category	Probability (P)	Impact (in Rs.)	Risk Exposure (RE)	Risk Response
R-001	**Low User Adoption:** Users (students) ignore the app and stick to existing tools	Business	0.6	Rs. 50,000	**Rs. 30,000**	**Mitigate**
R-002	**Performance Bottleneck:** The system fails to support the 1,000 concurrent user requirement.	Technical	0.4	Rs. 40,000	**Rs. 16,000**	**Mitigate**
R-003	**Security Breach:** Failure to prevent XSS/SQL Injection or encrypt data leads to a user data compromise.	Technical	0.2	Rs. 75,000	**Rs. 15,000**	**Mitigate**
R-004	**Key Developer Unavailable:** The Spring Boot or Firebase expert leaves the project, causing major delays.	Project	0.3	Rs. 45,000	**Rs. 13,500**	**Mitigate**
R-005	**Admin Inactivity:** The Admin user (faculty/staff) fails to post events, making the calendar useless.	Project	0.4	Rs. 30,000	**Rs. 12,000**	**Mitigate**

R-006	**3rd-Party Dependency Failure:** Firebase services (Auth or Database) have an outage, disabling login or chat.	Technical	0.3	Rs. 40,000	**Rs. 12,000**	**Mitigate**
R-007	**Poor Mobile Experience:** The web app is not fully responsive and functions poorly on mobile browsers.	Technical	0.5	Rs. 20,000	**Rs. 10,000**	**Mitigate**
R-008	**Ambiguous Requirements:** Conflict between requirements (e.g., 1000 users vs. "TBD") causes rework.	Project	0.4	Rs. 25,000	**Rs. 10,000**	**Avoid**
R-009	**Technology Incompatibility:** The Firebase chat library has conflicts with the Java Spring Boot backend.	Technical	0.2	Rs. 40,000	**Rs. 8,000**	**Mitigate**
R-010	**Cloud Cost Overrun:** Firebase or cloud deployment costs (chat, file storage) exceed the project budget.	Business	0.2	Rs. 30,000	**Rs. 6,000**	**Monitor**

3. RMMM (Risk Mitigation, Monitoring, and Management) Plan

This plan details the strategy for the two highest-priority, project-specific risks identified in the table.

RMMM Information Sheet 1

Risk ID:	**R-001**
Risk:	**Low User Adoption**
Description:	Students (end-users) find the app unhelpful, not user-friendly, or redundant compared to existing habits (like using WhatsApp groups). They do not adopt the app, and the project fails to achieve its purpose.
Risk Mitigation (Avoidance):	- **Phase 2:** Conduct user interviews and surveys *before* implementation to confirm the features solve a real problem. - **Phase 3:** Focus on a key value proposition that existing tools lack (the single, official, Admin-controlled calendar).

	- **Phase 3:** Design the UI/UX to be extremely simple, fast, and intuitive—faster than opening a competing app.
Risk Monitoring:	- **Phase 5:** Launch a beta version to a small, diverse group of students for User Acceptance Testing. - **Phase 5:** Track key metrics: "Daily Active Users," "Messages Sent," and "Time in App." - **Trigger:** Set a "risk trigger": If less than 50% of beta testers log in more than twice in the first week, this risk is becoming a reality.
Risk Management (Contingency Plan):	- **If the risk trigger is hit:** Immediately conduct follow-up interviews with beta testers to find out *why* they are not using the app. - Be prepared to **pivot**: Based on feedback, deprioritize failing features (like chat) and focus all remaining effort on the one feature they find useful (like the calendar). - If no features are found to be valuable, recommend project termination to avoid further waste of resources.

RMMM Information Sheet 2

Risk ID:	**R-005**
Risk:	**Admin Inactivity**
Description:	The Admin user (e.g., a professor or department staff) is the only one who can post official events. If they find the system complex or are too busy, they will not post. The calendar, a core feature, will be empty, making the app useless.
Risk Mitigation (Avoidance):	- **Phase 3:** Involve the Admin user in the design of the "Create Event" feature to ensure it is simple and fast. - **Phase 4:** Make the Admin dashboard and event creation tool the most polished and bug-free part of the application. - **Phase 6:** Provide a 30-minute training session and a simple one-page quick-start guide for the Admin.

Risk Monitoring:	- **Phase 7:** The project manager will actively monitor the live application's calendar. - **Trigger:** Set a "risk trigger": If no new events are posted by any Admin for 7 consecutive days, this risk is becoming a reality. - Check in with the Admin user bi-weekly for the first month to solicit feedback.
Risk Management (Contingency Plan):	- **If the risk trigger is hit:** The project manager will personally meet with the Admin to identify the bottleneck. - **Contingency 1 (User Issue):** If the Admin is too busy, assign a "deputy admin" (e.g., a TA or student representative) with permissions to help manage events. - **Contingency 2 (Tool Issue):** If the tool is too complex, allocate an immediate "hotfix" sprint to fix the UI issues reported by the Admin.

CONCLUSION:

From this experiment, I have learned that risk analysis is a two-part process that is essential for moving a project from a place of uncertainty to a position of control.

First, we learned that creating the Risk Table is not just about listing fears; it's about quantifying them. The most valuable step was calculating the Risk Exposure (RE), which instantly transformed a long, overwhelming list of potential problems into a clear, prioritized hierarchy. Second, we learned that identifying a risk is useless without a plan of action. Creating the RMMM Plan was the most critical part of this experiment, as it forced me to have a concrete strategy for the top risks. It taught me that every major risk needs three sub-plans: a plan to Mitigate it (how to prevent it), a plan to Monitor it (what triggers to watch for), and a Management plan (the contingency if it happens anyway).