



**PROJECT MANAGEMENT PLAN TEMPLATE**  
**(Group No: 03)**

**PROJECT MANAGEMENT PLAN**  
**Insurance Management System**

NAME	ID
MAHAMODUL HASAN MAHADI	20-42768-1
MD. MAHABUB MORSHED IJAZ	20-43434-1
SADIA FERDOUS	20-42793-1
JAMIL AHAMMED	19-41686-3

**SUBMISSION DATE: 27-08-2023**



<b>Table of Contents</b>	<b>Page Number</b>
<b>1.0 INTRODUCTION</b>	03
<b>2.0 PROJECT MANAGEMENT APPROACH</b>	03
<b>3.0 PROJECT TITLE</b>	04
<b>4.0 JUSTIFICATION</b>	04
<b>5.0 OBJECTIVES AND PROJECT SCOPES</b>	04
<b>6.0 OVERVIEW OF THE PROJECT</b>	06
<b>7.0 STAKEHOLDERS ANALYSIS</b>	07
<b>8.0 MILESTONE LIST</b>	08
<b>9.0 PROCESS MODE TO BE FOLLOWED</b>	09
<b>10.0 WORK BREAKDOWN STRUCTURE</b>	10
<b>11.0 ESTIMATION</b>	12
<b>12.0 RESOURCE REQUIREMENTS</b>	14
<b>13.0 PROJECT SCHEDULE</b>	14
<b>14.0 DELIVERY PLAN</b>	15
<b>15.0 RISK ANALYSIS</b>	17
<b>16.0 QUALITY CONTROL PLAN</b>	19
<b>17.0 BUDGET</b>	20
<b>18.0 CONCLUSION</b>	20
<b>19.0 REFERENCES</b>	21





## 1.0 INTRODUCTION:

Insurance Management System is a web-based application that provides insurance companies with an accessible platform to manage their operations. This system utilizes the force of the web, permitting clients to utilize its highlights from any area and whenever with an internet browser. With seamless integration and online accessibility, the web-based insurance management system enhances productivity and collaboration for insurance businesses.

## 2.0 PROJECT MANAGEMENT APPROACH:

- ❖ **Roles and Authority:** Our project roles and authorities have been clearly defined to ensure effective collaboration. The roles include:
  - **Project Manager:** Mahamodul Hasan Mahadi
  - **Business Analysts:** Sadia Ferdous
  - **Domain Expert:** Md. Mahabub Morshed Ijaz
  - **UI/UX Designer:** Jamil Ahammed
  - **Developers:** Mahamodul Hasan Mahadi, Jamil Ahammed
  - **Quality Assurance:** Md. Mahabub Morshed Ijaz
- ❖ **Resource Allocation:** Resources for our project will be allocated as follows:
  - **Human Resources:** We need full-time commitment from team members and domain experts.
  - **Hardware:** We need all the computers, servers, and testing devices for development and testing.
  - **Software Licenses:** We need to extend all the necessary software licenses for development and testing environments.
  - **Tools:** We need to utilize all development tools, version control systems, and project management software.
  - **Financial Provisions:** We need budget allocation for our project-related expenses.
- ❖ **Collaboration and Communication:** For project success, effective collaboration and communication play a vital role:
  - **Regular Meetings:** We need weekly project team meetings to discuss progress, find out challenges, and plan for our next steps.
  - **Milestone Reviews:** It will help stakeholders to showcase progress and gather feedback.
  - **Status Update:** Daily email updates to stakeholders highlighting achievements and any risks that occur in upcoming milestones.
  - **Feedback Mechanism:** It will help each team member to share updates with stakeholders and gather some suggestions from stakeholders.



### **3.0 PROJECT TITLE: Insurance Management System**

#### **4.0 JUSTIFICATION:**

Our website will provide the safest and most trusted service for customers when they buy this from us. Our website will also help companies to maintain their work in an organized way and track records. Our website will also save customers valuable time. For the customer's query, there will be a chat box, and experts from the respective company will also solve this query. Customers can easily renew and cancel their insurance without delay through our website. On our website, we will provide Whole data centralized in one place. Customers need not go everywhere they will get all the info of all the products of all branches in one place.

#### **5.0 OBJECTIVES AND PROJECT SCOPES:**

Three types of users: 1. Admin  
2. Insurance Manager  
3. Customer

##### **Admin User Functionality:**

1. Can register to the system.
2. Can Login to the system.
3. Can control their profile information (edit-delete-view etc.)
4. Can Logout to the system.
5. Can verify the insurance manager.
6. Can verify customer profiles.
7. Can block an insurance manager.
8. Can block a customer for disobeying certain rules.
9. Can add new insurance.
10. Can remove an insurance.

##### **Insurance Manager User Functionality:**

1. Can register to the system.
2. Can Login to the system.
3. Can control their profile information (edit-delete-view etc.)
4. Can Logout to the system.
5. Can add insurance.
6. Can cancel insurance.
7. Can verify customer profiles.
8. Can remove a customer.
9. Can block a customer for disobeying certain rules.
10. Can renew insurance.
11. Can search customer details etc.



### **Customer Functionality:**

1. Can register to the system.
2. Can Login to the system.
3. Can control their profile information (edit-delete-view etc.)
4. Can Logout to the system.
5. Can buy insurance.
6. Can cancel insurance.
7. Can edit information.
8. Can ask for renew insurance.
9. Can ask for insurance money.
10. Can comment or provide review etc.

### **Objectives:**

To develop an insurance management system that efficiently serves the requirements of admin, insurance manager, and customer by providing them with essential functionalities and controls for registration, profile management, customer management, and insurance management-related operations.

### **Sub-Objectives:**

#### **❖ User Management:**

- Allow users to register for the system
- Offer a safe way to login for authenticated access
- Enable users to control their personal information, including editing, deleting, and viewing their profiles
- To ensure a secure session termination, implement a logout function

#### **❖ System Admin:**

- Provide admin with overall control & management of the system
- Ensure that the information of all users is protected & kept private
- Execute appropriate access controls & permissions for different user roles

#### **❖ Verification & Blocking:**

- Allow admin to verify both insurance managers & customer profiles to ensure accuracy
- Allow admin to block insurance managers for any privacy violations
- Allow admin to block customers who disobey certain rules



#### ❖ Customer Management:

- Allow insurance managers to verify customer profile
- Allow customers to purchase insurance policies based on their requirements
- Allow customer to cancel their insurance policies as needed
- Allow customers to request policy renewals
- Allow customers to provide feedback, reviews regarding the insurance services

#### ❖ Insurance Management:

- Allow admin & insurance managers both to add new insurance policies to the system
- Allow admin to remove invalid insurance policies from the system
- Allow insurance managers to efficiently handle policy cancellations based on customer request
- Allow insurance managers to renew insurance policies for customers request

## 6.0 OVERVIEW OF THE PROJECT:

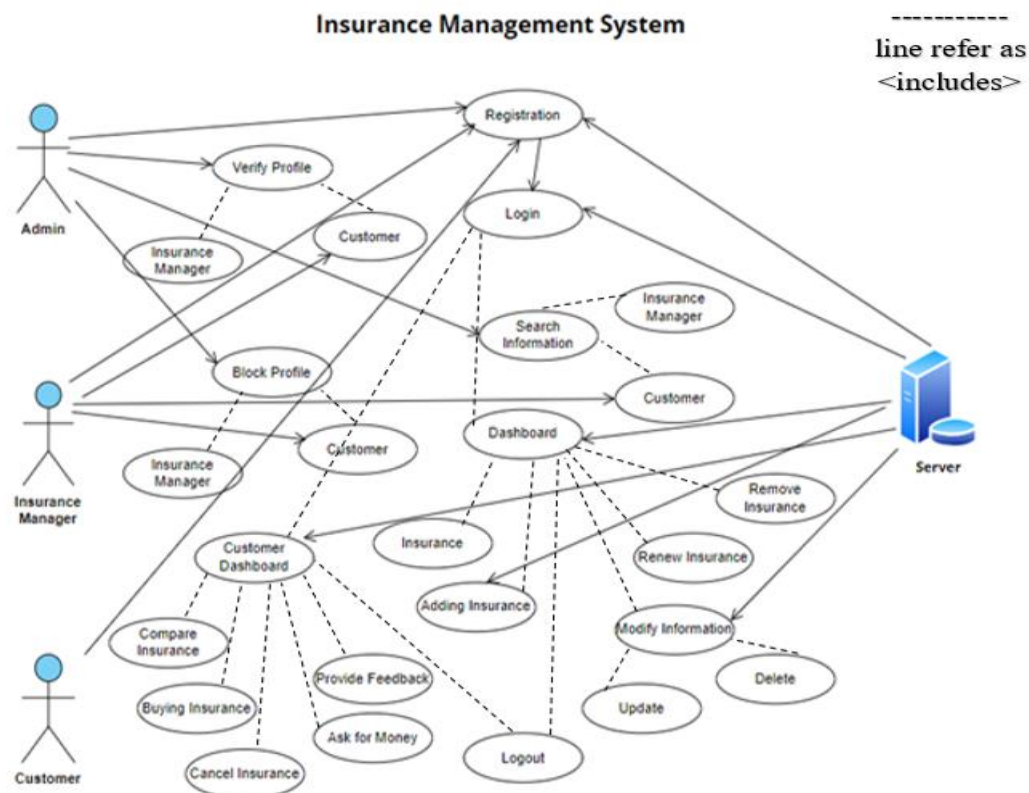


Figure 01: Use-Case Diagram of Insurance Management System.



This diagram shows the flow of use-case in an insurance management system. All three users can register, login, control their profile information and logout. All the information stored in a database server. The admin can verify insurance managers and customers, block them and add or remove insurance. The insurance manager can add and cancel insurance, verify and block customers, renew insurance and search for customer details. The customer can buy and cancel insurance, edit their information, ask for money and provide comments or reviews.

## **7.0 STAKEHOLDERS ANALYSIS:**

### **❖ Primary Stakeholders:**

- 1) Admin
- 2) Insurance Managers
- 3) Customers

### **❖ Secondary Stakeholders:**

- 1) Insurance Companies
- 2) Financial Institutions
- 3) IT Department
- 4) Data Analysis
- 5) Regulatory Authorities
- 6) Auditors
- 7) Third-Party Service Provider
- 8) Risk Management Consultants
- 9) Loss Adjusters
- 10) Medical Professionals
- 11) Business Partners



## 8.0 MILESTONE LIST

Milestone	Description	Date
Complete SRS	Meet with customers Identify needs and project constraints Product statement	4 weeks
Design	Functional Specification Technical Specification In-depth analysis of software requirements Define Scope, Objective Project Plan, Timeline, Resource Allocation Review Planning System UI, Database Schema idea generate High level UI design architecture Development of UI	20 weeks
Complete Coding	Define system functionality Define behavior Customer module Admin module Insurance module Research module functionality Integrate all module Research editing functionality Research grammar component Integrate database Integrate additional component	35 weeks
Complete Testing and Debugging	Functional and Non-functional testing System Testing Evaluate technical checking Evaluate all function	42 weeks
Deployment and Installation	Deploy the system in server User acceptance testing Customer Reviews	44 weeks

Table 01: Milestone list of Insurance Management System.





Figure 02: Timeline chart of Insurance Management System.

## 9.0 Process Model to be followed:

For our Insurance Management System, we have Chosen Waterfall model because of it might be beneficial for us. As we all know that it's a sequential and structured approach to software development, where each phase is completed before moving on to the next stage. Now here I'm going to explain why we are choosing the waterfall model for our project:

- **Well-Defined Phases:** In waterfall model, each phase has a clear focus, which can be advantages in complex projects like Insurance Management Systems. This structured approach can help us to ensure that all aspects of the system are thoroughly addressed.
- **Clear Requirements:** Because of we have a well-defined and stable set of requirements from the start of our project, this model can be effective.
- **Predictable Timeline:** In this model's sequential nature allows for better predictability in term of timeline and budget. Our project is under fintech domain and this type of project strictly need to follow the timeline and budget.



- **Internal and External Reporting:** Some fintech related company often need to provide progress reports to both internal stakeholders and external partners. Because of this model clear milestones and documentation can aid in producing these reports.
- **Well Defined Testing:** In this model, it typically includes dedicated testing phases after development. For our project, comprehensive testing is crucial to ensure both accurate policy calculations and security.
- **Stable Environment:** This model provides a stable environment it can be beneficial for us because all of our tools, technology, and working methodology are well-established.
- **Client Involvement:** this model encourages strong client involvement during the Requirement Gathering phase to Deployment phase.

## 10.0 WORK BREAKDOWN STRUCTURE:

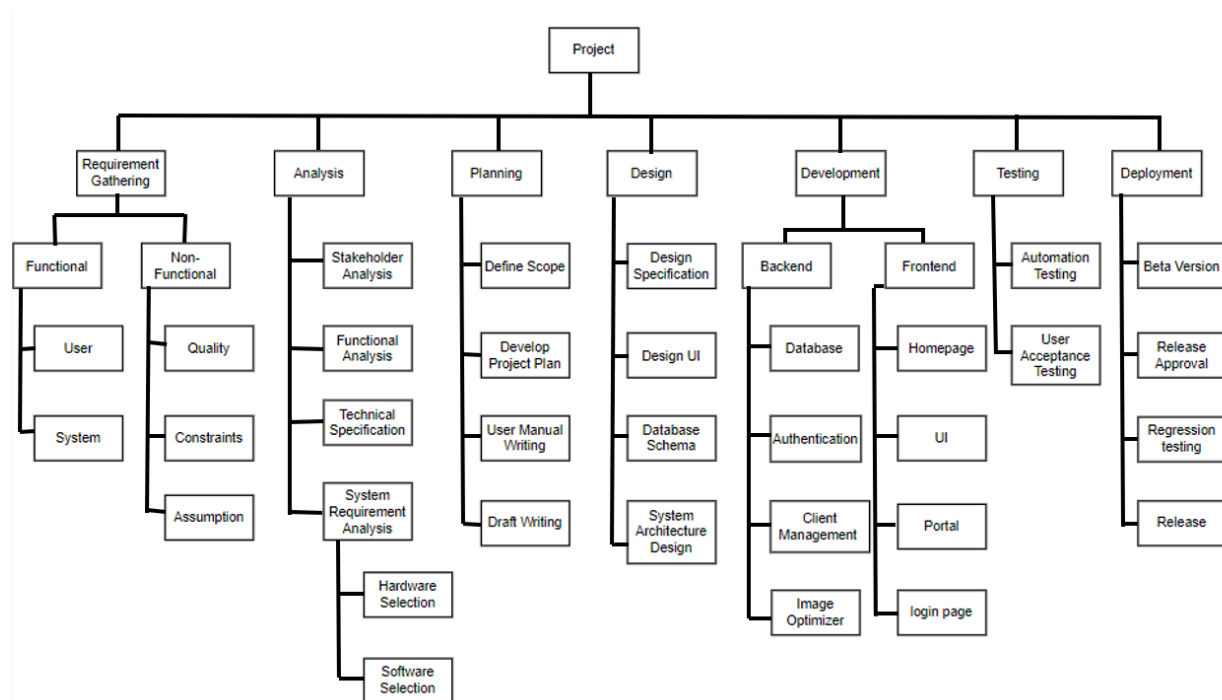


Figure 03: Work Breakdown Structure of Insurance Management System.



Activity	Duration (Weeks)	Precedence
A. Requirement Gathering	4	
B. Analysis	4	A
C. Planning	5	B
D. Hardware Selection	1	C
E. Software Selection	1	C
F. Design	7	C
G. File take on	5	C
H. Write User Manual	10	C
I. Development	15	F, D, E
J. Testing	7	I
K. Deployment	2	G, H, J

Table 02: Activity chart of Insurance Management System.

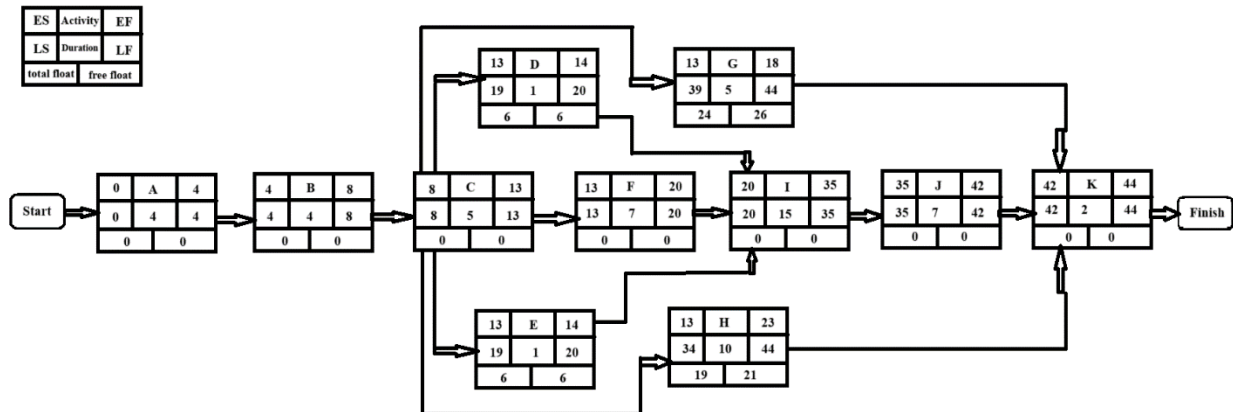


Figure 04: Activity diagram of Insurance Management System.

From this diagram we get:

- **Path 1:** A+B+C+F+I+J+K = (4+4+5+7+15+7+2) = 44 Weeks
- **Path 2:** A+B+C+D+I+J+K = (4+4+5+1+15+7+2) = 38 Weeks
- **Path 3:** A+B+C+E+I+J+K = (4+4+5+1+15+7+2) = 38 Weeks
- **Path 4:** A+B+C+G+K = (4+4+5+5+2) = 20 Weeks
- **Path 5:** A+B+C+H+K = (4+4+5+10+2) = 25 Weeks

After finding all the path we can say that **Path 1** is the Critical Path.



## 11.0: ESTIMATION:

### Effort Estimation

Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

### Constructive Cost Model:

Project Type: Organic

Co-efficient: 2.4 [a = 2.4, b = 1.05, c = 2.5, d = 0.38]

SLOC: 13000 Lines

Person Months, PM =  $(2.4 * 13^{1.05})$   
= 35.470

Development Time, DM =  $(2.5 * 35.470^{0.38})$   
= 9.70  
= 10 Months  
= 40 Weeks (22 Working days per month)  
= 1760 hours

Working Hour Required people, ST = PM/DM  
= 3.656  
= 4 people

### Budgeting:

#### Developer Salary for 10 Months:

Per Developer Salary Per Working Hour = 800 Taka  
Total Developer Salary =  $800 * 1760$   
= 14,08,000 Taka

#### Project Manager Salary for 10 Months:

Project Manager Per Month Salary = 60,000 Taka  
Project Manager Total Salary =  $60,000 * 10$   
= 6,00,000 Taka

#### Requirement Analysis:

Time Needed: 1 Month (22 Working Days = 176 Working Hour)  
Requirement Analysis Person's Hourly Wage = 600 Taka  
Total Requirement Analysis Expense =  $600 * 176$   
= 1,05,600 Taka



Transportation Cost Estimation: 25,000 Taka

Training And Hardware Expenses Estimation: 1,50,000 Taka

**Rent Expense:**

Room per month = 15,000 Taka

For 10 months =  $15,000 * 10$

= 1,50,000 Taka

**Total Utilities in 10 Months:** 50,000 Taka

**Maintenance (Till 8 Months After Delivery):**

Time For Maintenance Per Month = 8 Hours

Expense Per Hour = 1500

Total Estimated Time Needed for Maintenance = 64 hours

Total Estimated Maintenance Cost =  $1500 * 64$

= 96,000 Taka

**Marketing Expense:** 50000 Taka

**Web Hosting Expenses for 1 Year:** 5000 Taka

**Backup Storage Cost for 1 year:** 10000 Taka

**Consultant Expense:** 1,50,000 Taka

**Hardware Expense:** 5,00,000 Taka

**Total Estimated Expense:**  $14,08,000 + 1,05,600 + 1,50,000 + 1,50,000 + 5,00,000 +$

$1,50,000 + 25,000 + 96,000 + 50,000 + 5,000 + 10,000 +$

$50,000 + 6,00,000$

= 32,99,600 Taka

**Profit:**

25% Of Total Estimated Expense =  $32,99,600 * 25\%$

= 8,24,900 Taka

Total Estimated Expense =  $32,99,600 + 8,24,900$

= 41,24,500

= 41,50,000 Taka

**Total Project Budget: 41,50,000 Taka**



## 12.0 RESOURCE REQUIREMENTS

### 12.1 SOFTWARE REQUIREMENTS:

- Programming Languages: HTML, CSS, JavaScript (Front-end), PHP(Back-end).
- Web Frameworks: Laravel (PHP).
- Database: MySQL.

### 12.2 HARDWARE REQUIREMENTS:

- Memory: 4GB (Minimum)
- CPU: Intel Core i3 TM, 1.8GHZ
- GPU: Intel UHD Graphics 520

### 12.3 HUMAN RESOURCE REQUIREMENTS:

From the Constructive Cost Model:

- We have already calculated the number of people we need for our project.
- A total of four developers need to complete the project in a total of 44 weeks.

## 13.0 PROJECT SCHEDULE:

The development plan for the Insurance Management System will follow the Software Development Life Cycle (SDLC) and will consist of the following phases:

- **Requirement Gathering Phase:** In this phase is the phase where the process of systematically collecting, analyzing, and documenting the needs and expectations of stakeholders for a project. It involves interactions with users, clients, and subject matter experts to identify and define the features and functionalities of a system or software. The expected duration of this phase is four weeks.
- **Analysis Phase:** In this phase, the project team will analyze the requirements and gather the necessary data to develop the system. This will involve identifying the features and functionalities required, determining the data models, and designing the system architecture. The expected duration of this phase is four weeks.
- **Planning Phase:** In this phase, the project goals, objectives, and requirements will be identified. A feasibility study will also be conducted to determine if the project is viable and within the scope of the resources available. The expected duration of this phase is five weeks.



- **Design Phase:** In this phase, the project team will design the system based on the requirements and data gathered in the previous phase. This will involve creating wireframes, designing the user interface, and developing the database schema. The expected duration of this phase is seven weeks.
- **Development Phase:** In this phase, the actual coding of the system will take place. The system will be developed based on the design documents and specifications developed in the previous phase. The expected duration of this phase is fifteen weeks.
- **Testing Phase:** In this phase, the developed system will be tested to identify any defects and ensure that it meets the specified requirements. This will involve functional testing, performance testing, and security testing. The expected duration of this phase is seven weeks.
- **Deployment Phase:** In this phase, the system will be deployed in the production environment. The deployment will involve installing and configuring the software, migrating data, and conducting user training. The expected duration of this phase is two weeks.

The total expected duration of the project is 44 weeks. The project schedule will be as follows:

- **Week 1-4:** Requirement Gathering Phase
- **Week 5-8:** Analysis Phase
- **Week 9-13:** Planning Phase
- **Week 14-20:** Design Phase
- **Week 21-35:** Development Phase
- **Week 36-42:** Testing Phase
- **Week 43-44:** Deployment Phase

The project schedule may be adjusted depending on unforeseen circumstances, such as changes in requirements or unexpected delays. The project team will regularly monitor and update the project schedule to ensure that the project is completed within the specified timeline.



## 14.0 DELIVERY PLAN:

We're using the Waterfall Model for our Insurance Management System. Here is our project delivery plan:

- ❖ **Phase 1: Requirement Gathering (Week 1-4)**
  - Gather client requirements and expectations
  - Document detailed system requirements
  - Phase duration: 4 weeks
- ❖ **Phase 2: Analysis (Week 5-8)**
  - Finalize functional and technical specifications
  - Conduct in-depth analysis of system requirements
  - Phase duration: 4 weeks
- ❖ **Phase 3: Planning (Week 9-13)**
  - Define Project scope, objectives, and sub-objectives
  - Develop project plan, timeline, and resource allocation
  - Phase duration: 5 weeks
- ❖ **Phase 4: Design (Week 14-20)**
  - Design user interfaces and workflows
  - Develop detailed database schema and data models
  - Create high-level system architecture design
  - Phase duration: 7 weeks
- ❖ **Phase 5: Development (Week 21-35)**
  - Set up all the development tools and environments
  - Implement user authentication and access control
  - Develop policy management and claims processing modules
  - Integrate payment processing system with third-party
  - Phase duration: 15 weeks
- ❖ **Phase 6: Testing (Week 36-42)**
  - Creating test cases for each module
  - Conduct unit testing for each module
  - Perform integration testing of system components
  - Execute comprehensive system testing and ensure quality with QA
  - Phase duration: 7 weeks
- ❖ **Phase 7: Deployment (Week 43-44)**
  - Deploy the system to a production-like environment
  - Conduct final user acceptance testing with end-users
  - Phase duration: 2 weeks





## 15.0 RISK ANALYSIS:

ID	Description	Risk Categories	Probability	Cost	RE	Utility Policy
1	Human efforts	Development Environment	6	8	48	Resource certainty
2	Late changing on the requirement.	Project Size	5	8	40	Changing timely
3	Difficult privacy for data	Business Impact	6	10	60	Secured data
4	Low estimation of the size	Project Size	7	6	42	High estimation of the size
5	The system takes more time for some data	Project Size	8	6	48	The system takes less time to store and update data
6	Lack of Insurance receiver	Project Size	4	7	28	Have necessary customer
7	Lack of training in computer operations	Development Environment	6	5	30	Enough trained computer operations
8	High rate of staff turns over	Stuff Size	8	7	56	Low rate of stuff turns over

Table 03: Risk analysis chart of Insurance Management System.

In this table we take the probability and cost on a scale of 1 to 10. If the rating is higher than the more serious hazards, If the rating is lower than the risk is also less. Because of we have already completed identifying risk now we have to plan for our risk. There are some steps of planning where we will decide what to do with risks.



- ❖ **Risk acceptance:** There is no other way to accept the risk rather than prevent the risk.
  - **Example:** In the system testing it may show that the software detecting the problem of the Bank system but before publishing the problem system crashes sometimes but there is no time left to fix the problem. So, we need to accept the risk for the first release of our project and will try to develop it in another updated release.
  
- ❖ **Risk avoidance:** Avoiding the activity that could bring risk.
  - **Example:** We are considering 4 developers working in a team to build the project. One of the experienced team members could have had some problems and failed to complete the critical path activity in time. But involving another member who is eligible to handle the same work, we could involve him with the project from the starting period by maintaining the same or moderately high employee cost.
  
- ❖ **Risk transfer:** Transferring the risk prevention responsibilities to another team or organization.
  - **Example:** We have identified the risk that our system will not sense the detailed information about the insurance officer and the consumer if the system does not transfer data between different modules and our developers may fail to solve the problem. So, we can transfer the risk prevention responsibilities to other organizations in the basis of a contract.
  
- ❖ **Risk Reduction:** The actions to reduce a particular risk.
  - **Example for Project:** We have discussed the risk of the system crash in the risk acceptance steps. We could have developed a way that will save the data of unseen results so that the system doesn't need to take the data from the very first even though crashing the system.
  
- ❖ **Risk Mitigation:** Trying to reduce the post-impact of a risk.
  - **Example for our Project:** We could have a high rating of staff turnover and, in this situation, we can hire some excellent fresh graduates with enough salaries until the project completion time. Sometimes, there is confusion among the whole team about risk prevention or acceptance. To solve this confusion, there is a term called "Risk Reduction Leverage."



We can save 0.4 percent of this loss by paying BDT 50,000 to teach some individuals on-site work, but there are no end-user trainers. Thus, there is a 5% possibility of failures of BDT 1,20,000. We'll appreciate the RRL's value if it costs more than one.

$$\begin{aligned} \text{RRL} &= \{(\text{RE}_{\text{before}} - \text{RE}_{\text{after}}) / \text{Risk reduction cost}\} \\ &= \{(5\% \text{ of } 1,20,000) - (0.4\% \text{ of } 1,20,000)\} / 50,000 \\ &= (6000 - 480) / 50000 \\ &= (5520 / 50000) \\ &= 0.1104 \end{aligned}$$

As the  $\text{RRL} < 1$ , the step is not worth doing. These are the possible risks and prevention for our Insurance Management System project.

## 16.0 QUALITY CONTROL PLAN:

Here are some key points from a QA perspective:

1. Quality goals and objectives
2. Quality standards and guidelines
3. Stakeholder expectations
4. Documentation and Reporting
5. Load testing
6. Security and privacy testing
7. Database testing
8. UI / UX testing
9. API testing
10. Automation testing
11. Defect Management
12. Change Management
13. Continuous Improvement
14. Risk management
15. Schedule and milestones
16. Configuration management
17. Communication Plan
18. Lesson learned and knowledge sharing
19. Review and Approval Process
20. Exit Criteria for phases





## 17.0 BUDGET:

Our fee for the whole project, from the first step to the last one, will be 34,00,000 BDT. The breakdown is provided below:

Expenses	Total Amount (BDT)
Development cost	14,08,000
Project Manager cost	6,00,000
Requirement analysis	1,05,000
Transportation	25,000
Training	1,50,000
Office Rents	1,50,000
Utilities	50,000
Maintenance	96,000
Marketing	50,000
Web Hosting	5,000
Backup Storage	10,000
Consultant Expense	1,50,000
Hardware Expense	5,00,000
<b>Total Cost is</b>	<b>32,99,600</b>
For Profit 25% of total cost	8,24,900
<b>Total Budget is</b>	<b>41,50,000</b>

Table 04: Budget of Insurance Management System.

## 18.0 CONCLUSION:

To summarize, the Insurance Management System carries a new viewpoint to the insurance scene. Through cautious preparation, we've coordinated undertakings and obligations, making protection tasks more straightforward. By utilizing devices like action outlines, planning, and chance evaluations, we've guaranteed the undertaking's positive outcome. This system resembles an efficient toolkit for protecting the board, working on processes, and supporting effectiveness. It's intended to make protection tasks smoother, easier to use, and versatile to evolving needs. Eventually, this undertaking is set to make the protection of the executives a more smoothed out and effective process.



## 19.0 References:

1. Software Engineering Course Slides.
2. Object Oriented Analysis and Design Course slides.
3. Software Development Project Management Course Slides.
4. Visual Paradigm Online for drawing Use-Case Diagram.

