Internet Banking SysteM

PROJECT MANAGEMENT PLAN

**Revision record**

|  |  |  |  |
| --- | --- | --- | --- |
| version | Date | owner | description |
| V0 | 18/3/2024 | Aya Mostafa | Initial draft |
| V1 | 30/3/2024 | Sondos abdelgyed | Release strategy added |
|  |  |  |  |
|  |  |  |  |

TABLE OF CONTENTS

[**1 INTRODUCTION 1**](#_heading=h.1fob9te)

**Purpose of the Project Management Plan**

[**2 SCOPE MANAGEMENT 2**](#_heading=)

[**3 RESPONSIBILITIES 2**](#_heading=h.7ykpeesk0erd)

[**4 COMMUNICATIONS MANAGEMENT 2**](#_heading=)

[**5 CHANGE MANAGEMENT 2**](#_heading=h.le2bpxsuiouw)

[**6.**](#_heading=h.48rypluvipro) **PROBLEM** [**MANAGEMENT**](#_heading=h.le2bpxsuiouw) [**3**](#_heading=h.48rypluvipro)

[**6.1. Problem Discovery 3**](#_heading=h.6juamm2bu0ei)

**6**[**.1.1. Sources of Issue Discovery 3**](#_heading=h.ocfnshujzo5k)

**6**[**.1.2. Roles and Responsibilities 3**](#_heading=h.4iect5envd3s)

[**6.2. Issue Recording 3**](#_heading=h.58sz4rcc4x7p)

[**6.2.1. Issue Tracking System 3**](#_heading=h.gjcwx7hyzutd)

[**6.2.2. Issue Categories 4**](#_heading=h.qn7b0zf65l9r)

[**6.3. Issue Tracking and Prioritization 4**](#_heading=h.rh6c2ri758lo)

[**6.3.1. Issue Workflow 4**](#_heading=h.x0aoehn4lpyy)

[**6.3.2. Issue Prioritization Criteria 4**](#_heading=h.mc8h5ad5kipm)

[**6.4. Communication and Reporting 4**](#_heading=h.pkwr2ky9fnme)

[**6.5. Continuous Improvement 4**](#_heading=h.a0scxsgyfb60)

[**6.6. Escalation Process 4**](#_heading=h.701w52rky1ro)

**6.7. Issue log** [**5**](#_heading=)

**7.CONFIGURATION MANAGEMENT 5**

**7.1 Folder Structure 5**

**7.2 Naming Convention 5**

**7.3**[**.**](#_heading=h.701w52rky1ro) **Configuration Management List(CIL) 5**

**7.4. Roles and Responsibilities 6**

**8.TOOLS 6**

**9. REVIEWS**

# INTRODUCTION

## PURPOSE OF PROJECT MANAGEMENT PLAN

**The intended audience of the Internet Banking System PMP is all project stakeholders including the project customer, senior leadership, and the project team.**

**The purpose is to define the project scope, and team communication, define the possible problems & risks and how to handle them, the CMP structure, the tools used, responsibilities, and reviews.**

# SCOPE MANAGEMENT

**Is to provide a Web-based system for banking services where the client can easily perform a sequence of activities such as logging securely to perform inter-account transactions or with different accounts as well as viewing previous transactions, along with admin activities to monitor the banking services provided to the user.**

# responsibilities

* **plan:**

**project management role assigned for Aya**

**The team will share their knowledge and skills also.**

* **design:**

**the designing tasks will be assigned to the whole team**

* **development:**

**back-end development will be assigned to Sondos, Sohailla, and Mayar.**

**front-end development will be assigned to Aya, Sara, and Dina.**

* **testing:**

**will be assigned to the whole team every level of test will be assigned to a member**

# COMMUNICATIONS MANAGEMENT

* **regular face-to-face meetings on Saturdays**
* **the rest of the week are online meetings when needed**
* **meeting with the customer when needed either online or offline**

# Change MANAGEMENT

**Changes to the project upon the customer’s request will need to be taken into consideration as the highest priority to reach the customer’s satisfaction. The project leader will be in charge of dealing with any customer’s request. On receiving the change request, The project leader along with the team will respond immediately and start to discuss with the customer the consequences of the change, in addition to if the change can be applicable within the current release or it can be delayed to the upcoming one. Upon agreement with the customer, the status of the request will be announced [open-agreed-deferred.**

## 6. PROBLEM MANAGEMENT

**The Problem Management Plan outlines the procedures and guidelines for identifying, recording, tracking, and prioritizing issues related to the Internet Banking System project.**

**Effective problem management ensures timely resolution of issues and minimizes their impact on project deliverables.**

## 6.1. PROBLEM DISCOVERY

### 10.1.1. Sources of Issue Discovery

* **User Feedback: Gather feedback from end-users, stakeholders, and customer support channels.**
* **Testing and Quality Assurance: Identify issues during testing phases (functional, integration, and user acceptance testing).**
* **Monitoring and Logging: Set up monitoring tools to detect system errors, performance bottlenecks, and security vulnerabilities.**
* **Incident Reports: Document incidents reported by users or system alerts.**

### 10.1.2. Roles and Responsibilities

* **Project Manager: Oversees the problem management process.**
* **Development Team: Responsible for identifying and reporting issues.**
* **Quality Assurance Team: Validates reported issues and performs root cause analysis.**

## 6.2. ISSUE RECORDING

### 6.2.1. Issue Tracking System

* **Use a dedicated issue-tracking system (e.g., JIRA, Trello, or a custom-built tool) to record and manage issues.**
* **Include the following information for each issue:**
  + **Issue ID: Unique identifier.**
  + **Description: Detailed description of the issue.**
  + **Reporter: Name of the person reporting the issue.**
  + **Date Reported: Date when the issue was discovered.**
  + **Priority: High, medium, or low.**
  + **Status: Open, in progress, resolved, or closed.**
  + **Assigned To: Developer or team member responsible for resolving the issue.**
  + **Attachments: Screenshots, logs, or relevant files.**

### 

### 

### 

### 6.2.2. Issue Categories

* **Categorize issues based on their nature (e.g., functionality, performance, security, usability).**
* **Assign appropriate labels or tags to facilitate tracking.**

## 6.3. ISSUE TRACKING AND PRIORITIZATION

### 6.3.1. Issue Workflow

1. **Issue Reporting: Developers or end-users report issues.**
2. **Issue Validation: QA validates reported issues.**
3. **Root Cause Analysis: Investigate the underlying cause of the issue.**
4. **Issue Prioritization: Prioritize issues based on impact, severity, and urgency.**
5. **Assigning and Tracking: Assign issues to developers and track progress.**
6. **Resolution: Developers fix the issues.**
7. **Testing and Verification: QA verifies the resolution.**
8. **Closure: Close the issue once verified.**

### 6.3.2. Issue Prioritization Criteria

* **Impact: How severely the issue affects users or system functionality.**
* **Urgency: How quickly the issue needs resolution.**
* **Severity: The seriousness of the issue (critical, major, minor).**
* **Business Impact: Impact on business operations or customer satisfaction.**

## 6.4. COMMUNICATION AND REPORTING

* **Regularly update stakeholders on issue status, progress, and resolution.**
* **Use dashboards or reports to visualize issue trends and performance.**

## 6.5. CONTINUOUS IMPROVEMENT

* **Conduct post-implementation reviews to identify process improvements.**
* **Learn from resolved issues to prevent recurrence.**

## 6.6. ESCALATION PROCESS

* **Define escalation paths for critical issues that require immediate attention.**

## 6.7.ISSUE LOG

[Issue Log for Internet Banking System](https://docs.google.com/spreadsheets/d/1hsT1Nkh1fV3TpNDGyg2fdWmgypgjs6-HULaCuUfVJBg/edit?usp=sharing)

**7. CONFIGURATION MANAGEMENT**

**7.1 Folder Structure**

* **Project Management: Contains project management documentation, plans and requirements.**
* **Requirement: Contains requirements documentation specifying the functional and non-functional requirements of the system.**
* **Design: Contains design specifications and architecture documents.**
* **Development:**
  + **BackEnd: Contains backend code files and scripts.**
  + **FrontEnd: Contains frontend code files, including HTML, CSS, and JavaScript.**
  + **Database: Contains database schema files and scripts.**
* **Testing:**
  + **Test Design and Execution: Contains test plans, test cases, and test execution reports.**
  + **Bug Reports: Contains reports of identified bugs and issues.**

### 7.2 Naming Convention

#### Guidelines:

* **Prefix: Begin with a concise term indicating the category or type of component.**
  + **"REQ" for requirements.**
  + **"DES" for design elements.**
  + **"DEV" for development artifacts.**
  + **"TEST" for testing-related items.**
* **Descriptor: Include a brief, descriptive term indicating the purpose or function of the component.**
* **Identifier: Assign a unique identifier or name to each component.**
* **Delimiter: Use an underscore(\_) as a separator between the prefix, descriptor, and identifier.**

#### Example:

* **REQ\_User\_Login: Represents a requirement specification for user login functionality.**
* **DES\_UI\_Homepage: Refers to a design document outlining the user interface of the homepage.**
* **DEV\_Login: Denotes a development module responsible for user authentication.**
* **DB\_Customers\_Table: Represents a database table storing customer information.**
* **TEST-Integration-01-Plan: Refers to a test plan specifically for integration testing.**

**7.3** [**Configuration Item List**](https://docs.google.com/spreadsheets/u/0/d/10Im0UUe-v6zn_i_o-EYPzlo3YVcZXo4dgFzH269soQk/edit) **(CIL)**

**Contents of CI List:**

* **File Name: The unique identifier assigned to each configuration item.**
* **Link to Item: A reference or link to the actual item or document on GitHub Repo.**
* **Path: The location or directory where the item is stored within the project's file structure.**
* **Owner: The individual or team responsible for the configuration item.**

**Sample CI List:**

* **File Name: REQ-User-Login.docx**
  + **Link to Item: [Link to Requirement Document]**
  + **Path: /Requirement/Functional/**
  + **Owner: ABC**

**7.4 Roles and Responsibilities:**

* **Configuration Manager: Responsible for overseeing the Configuration Identification process and maintaining the Configuration Item List (CIL).**
* **Project Team: Collaborates in identifying configuration items and adhering to naming conventions when naming project components.**

**8- TOOLS**

* **THE APPLICATION WILL BE IMPLEMENTED IN:**

**1- Visual Studio Code.**

**2-intellij**

* **CONFIGURATION MANAGEMENT TOOLS:**

**1-Git** [**2-GitHub**](https://github.com/sohilaabdallaa/Internal-Banking-System)

* **TIME PLANNING AND MANAGEMENT TOOLS:**

[**1- trello**](https://trello.com/b/iZOzyn2f/internet-banking-system)

**9-REVIEWS**

**INITIAL REVIEWS**

* **At the beginning of each release(day 1 of the release) we will review our sprint tasks first and discuss any problem or any enhancements we can add .**
* **We here define our goals and our milestone.**
* **Make sure all the team members understand their roles and the sprint objectives.**

**TASK REVIEW**

**When a team member finishes his task he will send a confirmation e-mail to the project manager and add his peering colleague (in the CC section) and add the link attached to the newly added part( in the internal review phase branch ), the peering colleague will do the review for the task , then if this part pass from the review then he will send an email back to the project manager and his colleague and the team member who is responsible for the integration phase (in the CC section),if not then the peering colleague will send an e-mail also to the project manager with his colleague (in the CC section) and mentioned on the body of the e-mail what is the problem(repeat this process until this part pass the review phase ) .**

**After any situation of this task review , the reviewer will use this template to mention what is the status and the comments about this task** [**Task report sheet**](https://docs.google.com/spreadsheets/d/1Pl1s4sNWm7jnAxN0r538LIJAJKUBTY3EFwKktQZAgTQ/edit#gid=0)**.**

**COACHING REVIEW:**

**Every Tuesday, we will send an e-mail as a confirmation included the last version link from the configuration management tool and we will then check the comments of each release using this sheet:** [**Coaching review sheet**](https://docs.google.com/spreadsheets/d/1alEdP8X27mzFkDJsNED2nqczd_lIpsG6t6RemgbOn7Q/edit?usp=sharing)**.**

**RELEASE REVIEW**

**Preparation:**

* **First we will schedule a meeting**
* **Gather all relevant documentation, including the deliverables, release notes, test reports and any other pertinent information.**
* **Ensure that the deliverables have undergone thorough testing and meet the acceptance criteria defined for the project.**

**Agenda Setting:**

* **Overview of the deliverables included in the release.**
* **Summary of features, enhancements, or bug fixes implemented.**
* **Test results and quality assurance findings.**
* **Any known issues or risks associated with the release.**

**Review Process:**

* **Begin the release review meeting by providing an overview of the deliverables.**
* **Review each deliverable in detail, ensuring that they meet the project requirements and acceptance criteria.**
* **Discuss any issues, defects, or concerns identified during testing and verify that they have been adequately addressed.**

**Decision Making:**

* **Based on the review discussions and findings, decide on whether the deliverables are approved for release.**
* **If there are any outstanding issues or concerns that need to be addressed before release, determine the appropriate actions and assign responsibilities for resolution.**

**Release Strategy :**

* **Communication Plan :**

**Stakeholders:**

**Communication will be through the Mail where the project leader will inform the stakeholders with the status of the project.**

**Through the mail the project leader will arrange a meeting timing to be conducted at the middle of each release and by the end of each release .**

**After the end of each a meeting a mail will be sent to keep them with the updates that took place.**

* **Release Time : Release time will take place weekly .**
* **Name Convention of the release : <Realease\_Number of the release >**
* **Methods of delivery:**

**Delivery will be through the mail .**

* **Time Line Schedule:**

**Tracking the team’s tasks will be through** [**Trello**](https://trello.com/b/iZOzyn2f/internet-banking-system)