**FUNDAMENTAL BANKING SYSTEM**

**DELIVERABLE-1 PROJECT PLAN**

**09/07/2020**

**By**: **TECH CODERS**

**VERSION HISTORY:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented By** | **Revision Date** | **Approved By** | **Approval Date** | **Reason** |
| 1.0 | *SAGAR*  *TULLURU* | *09/09/2019* | *TEAM* | *09/9/2019* | *Initial draft* |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**TABLE OF CONTENTS**

1. [PROJECT DESCRIPTION.. 4](#_TOC_250000)
   1. [**Tasks and Sub-Tasks Involved in application………………………………..4**](#_bookmark0)
2. **[TIMELINE………………………………………………………………………………...5](#_bookmark1)**
3. **[RISK MANAGEMENT](#_bookmark1)…………………………………………………………………..6**

**3.1 Risk Identification……………………………………………………………..6**

**3.2 Risk Analysis…………………………………………………………………...6**

**3.3 Risk Response Planning……………………………………………………….6**

**3.4 Risk monitoring and control………………………………………………….7**

1. **ROLES & RESPONSIBILITIES………………………………………………………...7**
2. **DELIVERABLE-CONTRIBUTION………………………………………………….....8**

**Figures:**

**Figure 1:** Project timeline using Gantt Chart

**Tables:**

**Table 1:** Probability and Impact of Risks

**Table 2:** Delivery-1 Contributions

1. **PROJECT DESCRIPTION**

The “Fundamental Banking System” is an internal web application developed for banking authorities (For Manager and Cashier). The application authenticates the authorities and redirect them to their respective pages. This will be web-based application. The main functionalities of the application will be Login, Customer Management, Account Management, Deposit, Withdraw, Transfer and Mini Statement. The application will be developed using HTML, CSS, JavaScript, jQuery, C#, Microsoft Visual Studio, MVC template, SQL Server and Microsoft SQL Server Management Studio. The estimated duration of the project will be 3 months (one semester) ending in December 2020.

**1.1 Tasks and Sub-Tasks Involved in application:**

**Home Page**

1. Logo and Title
2. Navigation Bar
3. Slider
4. About us
5. FAQ

**Login**

1. Design a Login page
2. Encryption

**Customer Management:**

1. Add New Customer
2. List Customer
3. Search Customer
4. Update Customer
5. Delete Customer

**Account Management:**

1. Add Account
2. List Account
3. Search Account
4. Update Account
5. Delete Account

**Cashier Services:**

1. Deposit
2. Withdraw
3. Transfer
4. Mini Statement
5. **TIMELINE**

As a part of the project, a high-level planning for timeline is scheduled by using a Gantt chart. These requirements may be changed accordingly as the time progresses.

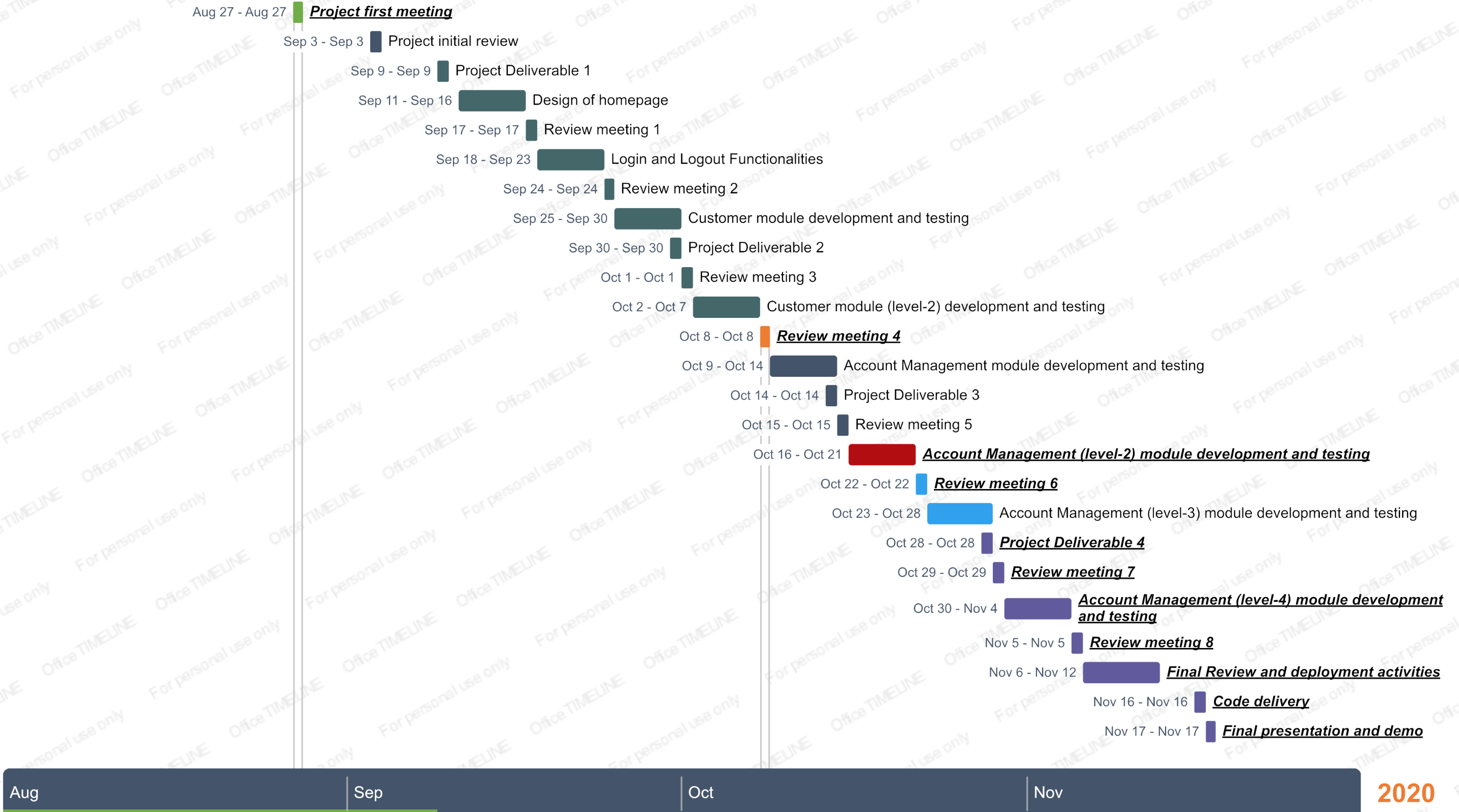


Figure 1

1. **RISK MANAGEMENT**

Risk management involves four major phases which are as follows:

1. Risk identification
2. Risk analysis
3. Risk response planning
4. Risk monitoring and control.

**3.1 Risk Identification:**

1. **Application Security:** As we are trying to develop an internal software application for fundamental banking. Application Security can pose a potential threat for us.
2. **Exception Handling:** We always thrive to make our functionality work, but we might not be sure about the unexpected exceptions that comes our way. So, Unexpected exceptions will be another threat we need to consider.
3. **Database Inconsistencies:** In any application, data flows between application and database. We must make sure that there will be no inconsistencies in saving or retrieving data to/from database. These inconsistencies can possibly have a high impact.

**3.2 Risk Analysis:**

In this, we prioritize the risks according to their probability and impact on the ongoing project.

|  |  |  |
| --- | --- | --- |
| **Risk** | **Probability** | **Impact** |
| Application Security | High | High |
| Exception Handling | Medium | High |
| Database Inconsistencies | Low | High |

Table 1

* High - An event which is very likely to happen and can impact the project severely.
* Medium - An event which has 50-50 chance of occurring and can cause noticeable impact.
* Low - An event that is unlikely to occur and can cause low impact on project.

**3.3 Risk Response Planning:**

* **Application Security:**

1. Restricting Unauthorized access
2. Encrypting the DB connection string
3. Encrypting the passwords in the login.

* **Unexpected Exceptions:**

We will deal the unexpected exception by implementing Exception Handling (try, catch and finally blocks) for every functionality.

* **Database Inconsistencies:**

These Inconsistencies may happen due to DB connection issues or any mismatch between the DB structure and application code. These can be avoided by proper planning while creating DB and Integration of DB to application.

**3.4 Risk monitoring and control:**

* **Monitoring:**

Assess the currently defined risks on regular basis by establishing periodic reviews. We communicate risk management status and risk response advancements in weekly meetings. Therefore, we can keep track of all the risks and can plan their remedies accordingly.

* **Reevaluation:**

We validate previous risk assessments to assess their probability and impact again by evaluating the effectiveness of actions taken. Identify any new risks which can be added to the risk registry.

* **Contingency Plans:**

1. We will implement all the risk response plans to mitigate all the risks.
2. We try to validate all the risk mitigation strategies and alternatives.
3. Take the corrective actions when risk occurs.
4. Assess the impact on the project of actions taken.
5. Ensure the project plan has been maintained.
6. **ROLES& RESPONSIBILITIES**
7. **Vishnu Theja Reddy Madavaram Janaki**

Role: Database Administrator, Testing, Documentation

Responsibilities:

* Creating the database with all the required tables, views and stored procedures.
* Testing the database to ensure accurate data storage and retrieving into the database.
* Documenting the accomplished work.

1. **Rushi Savulge**

Role: UI/UX Developer, Testing, Documentation

Responsibilities:

* Developing the front-end code for web pages.
* Testing the developed code and identifying the defects or bugs in the web pages.
* Documenting the accomplished work.

1. **Akhil Totakura**

Role: C# Developer, Testing, Documentation

Responsibilities:

* Developing the internal functionalities code.
* Testing the code for the modules developed.
* Documenting the accomplished work.

1. **Gunavardhan Reddy Jakkidi**

Role: UI/UX Developer, Testing, Documentation.

Responsibilities:

* Developing the front-end code for web pages.
* Testing the developed code and identifying the defects or bugs in the web pages.
* Documenting the accomplished work.

1. **Sagar Tulluru**

Role: C# Developer, Testing, Documentation

Responsibilities:

* Developing the internal functionalities code.
* Testing the code for the modules developed.
* Documenting the accomplished work.

Above roles are assigned according to the initial plan of developing the application. As time evolves, all the team members may work collaboratively on a single module based on the circumstances. These above roles do not restrict them to their particular task, any team member can chip in other teammate work so that work doesn’t get delayed and making sure project goes in perfect pace.

1. **DELIVERABLE-1 CONTRIBUTION**

Member contribution table (should describe who wrote what parts of the report)

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution(%) | Note (if applicable) |
| **Vishnu Theja Reddy Madavaram Janaki** | Worked on 1,2,3,4 points of the report. | 20% |  |
| **Rushi Savulge** | Worked on 8,9 points of the report. | 20% |  |
| **Akhil Totakura** | Worked on point 7 of the report. | 20% |  |
| **Gunavardhan Reddy Jakkidi** | Worked on point 6 of the report and PPT for presentation. | 20% |  |
| **Sagar Tulluru** | Worked on 5,6 points of the report. | 20% |  |

Table 2