UE21CS341A - Software Engineering

**PROJECT PLAN DOCUMENT**

**ATM SIMULATOR**

Team #: \_\_T\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | PES1UG21CS652 | Swaraj MK | | PES1UG21CS618 | Srikrishna B | | |  |  | | --- | --- | | PES1UG21CS662 | Tanmay Praveen Udupa | | PES1UG21CS659 | T P Shriambikesh | |

**Lifecycle**

Lifecycle to be followed:

For an ATM Simulator project, the Agile development lifecycle is a suitable choice due to the following reasons:

Iterative Development: Agile allows for iterative development, which is beneficial for a complex project like an ATM Simulator where requirements may evolve.

Customer Collaboration: Regular feedback and involvement of stakeholders ensure that the ATM Simulator aligns with user expectations and business needs.

Flexibility: Agile methodologies provide flexibility to adapt to changing requirements, allowing the team to respond to unforeseen challenges.

Continuous Improvement: The regular retrospective meetings encourage continuous improvement in processes and product quality.

Early and Regular Delivery: Agile promotes the early and regular delivery of increments, providing tangible value to users throughout the development process.

**Tools Used for this Project**

Planning Tools: Microsoft Excel

Design Tools: Draw.io, ERD Plus and Visual Paradigm

Development Tools: Flask, Python, sqlite

Version Control: Git Version Control

**Deliverables classified as reuse/build components**

Determine all the deliverables and categorize them as reusable/build components, and provide justification for each categorization

Deliverables classified as reuse/build components

User interface (UI) design (Reusable component)

The UI design is a core component of the ATM simulator and can be reused in other software projects, such as other banking applications or training simulators.

Account management module (Reusable component)

The account management module is a self-contained unit that can be easily integrated into other software systems that require user account management functionality.

Balance inquiry module (Reusable component)

The balance inquiry module is a simple and straightforward component that can be reused in other banking applications or financial data management systems.

Cash withdrawal module (Reusable component)

The cash withdrawal module is a critical component of the ATM simulator and can be reused in other banking applications or financial transaction processing systems.

Cash deposit module (Reusable component)

The cash deposit module is another essential component of the ATM simulator and can be reused in other banking applications or financial transaction processing systems.

Error handling module (Reusable component)

The error handling module is a crucial component that can be reused in various software systems to effectively handle and report errors.

**Work Breakdown Structure**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | |  | |  | | | |  | |
| **Milestone description** | **Category** | **Assigned to** | | **Progress** | | **Start** | | | | **Days** | |
| **Project Initiation** |  |  | |  | |  | | | |  | |
| Project Kick-Off | On Track | All Members | | **100%** | | 9/6/2023 | | | | 1 | |
| Identify Stakeholders | On Track | All Members | | **100%** | | 9/7/2023 | | | | 1 | |
| **Requirements Analysis** |  |  | |  | |  | | | |  | |
| Define Use Cases | Low Risk | Swaraj MK | | **100%** | | 9/10/2023 | | | | 1 | |
| Gather Requirements | Low Risk | Tanmay U | | **100%** | | 9/14/2023 | | | | 1 | |
| Prioritise Requirements | Med Risk | Srikrishna,Shriambhikesh | | **100%** | | 9/17/2023 | | | | 2 | |
| **System Design and Plan** |  |  | |  | |  | | | |  | |
| Architecture Design | Med Risk | tanmay U, Srikrishna | | **100%** | | 9/19/2023 | | | | 2 | |
| Database Design | Low Risk | Swaraj mk, Shriambhikesh | | **100%** | | 9/27/2023 | | | | 4 | |
| User Interface Design | Low Risk | All Members | | **100%** | | 9/28/2023 | | | | 5 | |
| Security Planning | High Risk | All Members | | **100%** | | 10/1/2023 | | | | 6 | |
| Infrastructure Planning | Low Risk | Shriambhikesh | | **100%** | | 10/6/2023 | | | | 2 | |
| **Development : 1) Frontend(first 3), 2) Backend(last 5)** | | |  | |  | |  |  |  | |
| Implement User Login | High Risk | All Members | | **100%** | | 10/7/2023 | | | | 2 | |
| Create user Interface | Med Risk | swaraj mk, Srikrishna | | **100%** | | 10/8/2023 | | | | 2 | |
| Implement transaction Proessing | Med Risk | tanmay U, Shriambhikesh | | **100%** | | 10/10/2023 | | | | 1 | |
| Implement User Database | Low Risk | tanmay U, Shriambhikesh | | **100%** | | 10/15/2023 | | | | 3 | |
| Configure User Database | Low Risk | swaraj mk, Srikrishna | | **100%** | | 10/16/2023 | | | | 2 | |
| Implement transaction Database | Med Risk | swaraj mk, Srikrishna | | **100%** | | 10/20/2023 | | | | 3 | |
| Configure transaction Database | Med Risk | All Members | | **100%** | | 10/22/2023 | | | | 5 | |
| Create views and routes | Med Risk | srikrishna | | **100%** | | 10/25/2023 | | | | 4 | |
| **Testing** |  |  | |  | |  | | | |  | |
| testing frontend | Low Risk | Shriambhikesh | | **100%** | | 11/2/2023 | | | | 2 | |
| testing backend | Low Risk | Tanmay | | **100%** | | 11/4/2023 | | | | 2 | |

**Effort Estimation (in person-months)**

(1 year = 260 working days, then 1 month = 260/12 = 21.66 working days. So 7 full working days for one person would be 7/21.66 = 0.323 person‑months.)

Project Initiation :

2 working days

2/21.66 = 0.0923 person-months

Requirement Analysis :

4 working days

4/21.66 = 0.184 person-months

System Design and Plan :

19 working days

19/21.66 = 0.877 person-months

Development(frontend and backend) :

22 working days

22/21.66 = 1.015 person-months

Testing :

4 working days

4/21.66 = 0.184 person-months

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Gantt Chart**



