

A Time Monitoring Tool

Requirements Definition and Business Modeling

Group 5

Team Members

Ratnesh Pawar
Mayank Sharma

Rohan Sonawane
FNU Sumit Kumar

Abhishek Dhale
Neil Duraiswami

INDEX

1. Requirements Definition	2
i. Introduction	2
ii. Requirements Analysis	3
iii. Requirement Elicitation	3
iv. Non-Functional Requirements	4
v. Functional Requirements	5
vi. User Stories	6
2. Business Process and Functional Modeling	9
i. Introduction	9
ii. Actors List	10
iii. Use case list and Use Case Diagram	11
iv. Activity Diagrams	13
v. Use Cases	17
vi. Verification & Validation	19

FIGURES LIST

- 1. Use Case Diagram for Time Management Tool**
- 2. Activity Diagram for Time Management Tool**
 - 2.1. Activity Diagram for User Authentication
 - 2.2. Activity Diagram for Weekly Timesheet
 - 2.3. Activity Diagram for Record Timestamp
 - 2.4. Activity Diagram for Activity Categorization
 - 2.5. Activity Diagram for Time Budgeting
 - 2.6. Activity Diagram for Manage Accounts
 - 2.7. Activity Diagram for Generate Reports
 - 2.8. Activity Diagram for Resource Tracking

PART 1: Requirements Definition

Introduction

The Time Monitoring Tool (TMT) aims to tackle a common challenge faced by software development companies: the need for dependable and uniform data regarding time allocation across various project activities. Accurate time data is crucial for monitoring project advancement, precise client invoicing, contractor payments, and the enhancement of future project planning.

The primary objective of TMT is to facilitate systematic and consistent tracking of where time and effort are invested in software development projects. Furthermore, it intends to enable project managers to compare actual resource utilization with initial projections, thereby enhancing the precision of future project planning and development.

The primary functions of Time Management Tools are:

- **Recording Timestamps:** Using a web UI interface, software developers may quickly record the start and end timestamps of their tasks.
- **Activity Categorization:** Activities will be of different types which will help the developer and project manager to differentiate between activities & help to generate reports for tracking various activities.
- **User Authentication:** To avoid any security breach and illegal actors accessing or entering data into the system and providing different functions to different types of users we use user authentication.
- **Data Analysis:** Project Managers can use the Time Management Tool to check timestamp records and develop Analytical Reports, Charts, and graphs to get hidden insights and make decisions.
- **Resource Tracking:** The system goes beyond time tracking and managing the assignment of workers to tasks or projects to give a full resource management solution.
- **Weekly Time Sheets Data:** This Data will be generated from the data entered by the Project developer and manager which helps them to get insights on their weekly work which will help them to prioritize tasks.
- **Prioritizing Tasks:** Allow users to assign tasks according to priority and manage their tasks as necessary. This helps us to deliver projects on time without any backlogs.
- **Time Budgeting:** Give project managers the ability to set time budgets for tasks or projects and receive notifications when resources are being overused or are running low.

Requirements Analysis:

The process of analyzing the requirements for the Time Monitoring Tool (TMT) project is a critical phase that lays the foundation for the project's success.

- **Stakeholder Survey:** We contacted important stakeholders such as project managers and developers to determine what they need from the TMT system.
- **Checked documentation:** We examined current time tracking and project management documentation and data to identify what might be improved.
- **Use-Case Analysis:** We created extensive use cases to understand how users (developers and project managers) interact with the TMT system.
- **Functional & Non-Functional Requirements:** We divided requirements into two categories: things the system should be able to accomplish functional and non-functional requirements.
- **Double-checked:** We double-checked the data from stakeholders and confirmed the requirements.
- **Tracking:** We built a traceability matrix to link requirements to their sources and enable thorough testing.
- **Set Priorities:** We prioritized requirements by significance, first addressing the most critical ones.

Requirement Elicitation:

Requirement elicitation is a crucial phase in the development of the Time Monitoring Tool (TMT) project, where the goal is to gather and refine the requirements effectively. List of techniques applied to achieve it:

- **Stakeholder Interviews:** Conducted 1-on-1 interviews with key stakeholders, including project managers, software developers, and end-users. These interviews allowed for in-depth discussions to uncover their needs, expectations, and pain points related to time tracking and project management.
- **Document Analysis:** Thoroughly reviewed existing documents such as project proposals, internal reports, and historical data related to time tracking and project management. This analysis provided insights into past practices and highlighted areas where improvement was needed.
- **Prototyping and Mock-ups:** Developed interactive prototypes and mock-ups of the TMT system's user interface. These prototypes were shared with stakeholders to gather feedback on the system's look and feel, leading to refined user interface requirements.
- **Observation and Job Shadowing:** Observed developers and project managers daily to gain firsthand insights into their time-tracking practices. Job shadowing helped identify implicit requirements and bottlenecks in the existing processes.
- **Competitive Analysis:** Conducted a competitive analysis of existing time tracking and project management tools in the industry. This benchmarking exercise identified features and functionalities that were considered best practices and desirable for the TMT system.

- **Requirement Analysis:** Leveraged AI-driven text analysis tools to process and analyze unstructured data from interviews, surveys, and documents. This AI-powered analysis uncovered hidden patterns and insights within the textual data, helping to refine and prioritize requirements.

Non-Functional Requirements

Non-functional requirements encompass the operational, performance, security, and cultural/political aspects of the TMT project. They serve as essential guidelines to ensure that the system operates effectively, performs optimally, maintains robust security, and respects cultural and political considerations in its design and implementation.

Operational Requirements:

1. **Availability and Uptime:**
 - The TMT system must be available 24/7, with planned maintenance windows communicated in advance.
 - Scheduled maintenance should occur during non-business hours to minimize disruption.
2. **Backup and Recovery:**
 - Regular automated backups of the database and system configuration must be performed.
 - The system should have a recovery time objective (RTO) of no more than 4 hours in case of unexpected outages.
3. **Logging and Monitoring:**
 - The system should maintain detailed logs of user activities, errors, and security-related events.
 - Real-time monitoring tools must be in place to detect and respond to system anomalies.
4. **Scalability:**
 - The architecture should support seamless scalability to accommodate growth in user and data volumes.
 - Load balancing mechanisms should be employed to distribute traffic efficiently.

Performance Requirements:

1. **Response Time:**
 - The system should respond to user interactions within 2 seconds for standard operations.
 - Complex operations, such as generating reports, may have extended response times but should not exceed 10 seconds.
2. **Throughput:**
 - The system should support concurrent access by at least 500 users without degradation in performance.
 - Peak usage periods should be accommodated without significant slowdowns.
3. **Resource Utilization:**

- The system should be optimized to minimize CPU, memory, and network bandwidth usage.
- Resource consumption should be proportional to the scale of user activity.

Security Requirements:**1. Authentication and Authorization:**

- Users must authenticate securely through a multi-factor authentication (MFA) mechanism.
- Role-based access control (RBAC) should be enforced to ensure that users can only access authorized data and functionality.

2. Data Encryption:

- Data at rest should be encrypted using strong encryption algorithms.
- Data transmitted between clients and the server should be encrypted using TLS.

3. Audit Trails:

- Comprehensive audit trails must be maintained, recording all user actions, including logins, data modifications, and access attempts.
- Audit logs should be tamper-evident and stored securely.

4. Vulnerability Management:

- Regular vulnerability assessments and penetration testing should be conducted.
- Identified vulnerabilities must be promptly patched or mitigated.

Cultural and Political Requirements:**1. Cultural Sensitivity:**

- The system should be designed to accommodate diverse cultures, languages, and regional differences.
- User interfaces and documentation should be available in multiple languages if applicable.

2. Data Privacy Compliance:

- The system must comply with international data privacy regulations, such as GDPR as relevant to the user base.

3. Political Neutrality:

- The system and its documentation should maintain political neutrality and avoid any bias or sensitive content that could lead to political or cultural conflicts.

4. Accessibility:

- The TMT system should adhere to accessibility standards (e.g., WCAG) to ensure it is usable by individuals with disabilities, respecting cultural diversity in accessibility needs.

Functional Requirements:

- **Recording Timestamps:** Software developers can log the beginning and end times of their activities using the web interface provided by the TMT system. Timestamps must contain information about the nature, duration, and relevant project/task association of the activity.
- **Activity Categorization:** To simplify organization and analysis, the system must categorize all recorded activities. Users, including programmers and project managers, shall be able to categorize activities and match them to the type of work being done using predefined or custom categories.
- **User Authentication:** To guarantee data security and the accuracy of timestamps, robust user authentication, and access control mechanisms must be put in place. Only authorized users, such as software developers and project managers, shall have access.
- **Data Analysis:** The TMT system's tools are available to project managers for in-depth analysis of timestamp records. To assist in project decision-making, the system must produce insightful reports, charts, and graphs that show how actual resource usage compares to estimates.
- **Resource Tracking:** The system will assign personnel to tasks or projects in addition to tracking time. It will offer resources for effectively managing resource allocation. This feature will be available to project managers and software developers.
- **Weekly Time Sheets:** The TMT system will make it possible to extract weekly time sheets, giving a glimpse into how developers spend their time each week. These time sheets must have activity breakdowns and be available to project managers and software developers.
- **Task Prioritization:** To effectively manage time and concentrate on high-priority work, users, including software developers and project managers, shall have the ability to assign priority levels to tasks and activities.
- **Time Budgeting:** Time budgets for tasks or projects may be set by project managers. When resource allocation falls short of or exceeds the allotted time, the system must send out notifications.

User Stories

User Story 1: Time Recording

As a developer, I want to be able to use my web browser So that I can easily record the time I spend on various activities during the day.

Acceptance Criteria:

- I can access the Time Monitoring Tool (TMT) via a web browser.
- I can log in securely using my credentials.
- I can select the type of activity (e.g., coding, testing) and enter the time duration.
- I can add a description or notes to provide context for the activity.
- I can save the timestamp record for future reference.

User Story 2: Time Measurement

As a developer, I want the TMT system to accurately measure and log the time for each activity So that I can have a precise record of how I spend my time.

Acceptance Criteria:

- When I start an activity, the TMT system starts tracking time.
- When I finish the activity, the system stops tracking time and records the duration.
- The system should display the total time spent on each activity.

User Story 3: Weekly Time Sheets

As a developer, I want to generate weekly time sheets from the TMT system So that I can report my time accurately to my manager.

Acceptance Criteria:

- I can select a specific week for which I want to generate a timesheet.
- The timesheet should list all the activities I recorded for that week, along with their durations.
- I can export the timesheet in a downloadable format (e.g., PDF or CSV).

User Story 4: Data Consolidation

As a project manager, I want to consolidate data for an entire project from the TMT system So that I can have a comprehensive view of resource allocation.

Acceptance Criteria:

- I can select a specific project and timeframe for data consolidation.
- The system should aggregate time records from multiple developers and activities for that project.
- The consolidated data should be presented in a clear and organized format for analysis.

User Story 5: Resource Tracking

As a project manager, I want to track the resources devoted to a project using the TMT system So that I can make informed decisions about resource allocation.

Acceptance Criteria:

- I can assign developers and teams to specific project phases or tasks.
- The system should track the time and effort each resource dedicates to the project.
- I can view resource allocation reports to assess resource usage.

User Story 6: Security and Authentication

As a user, I want the TMT system to enforce robust security measures So that I can trust that my timestamp records are safe and reliable.

Acceptance Criteria:

- The system should require secure user authentication, including multi-factor authentication (MFA).
- User data, including login credentials, should be securely stored and encrypted.

- Only authorized users should have access to timestamp records.

User Story 7: Reporting and Analysis

As a project manager, I want to use the reporting and analysis features of the TMT system So that I can make data-driven decisions about project planning and development.

Acceptance Criteria:

- I can generate reports that provide insights into resource allocation, including time spent on different activities.
- The system should allow me to compare actual versus estimated efforts for better project estimation.
- I can access visualizations and charts to facilitate data analysis.

These user stories capture the essence of each function in the Time Monitoring Tool (TMT) system and provide clear objectives from the perspective of different users, including developers and project managers. These user stories will serve as valuable guidance during the development process, ensuring that the system meets user expectations and needs.

Part II – Business Process and Functional Modeling

Introduction:

In modern software development, the efficient tracking of time spent on various project activities holds paramount importance. This report comprehensively explore the Time Monitoring Tool (TMT) project, illuminating its background and the overarching purpose it seeks to fulfill.

Background:

Software development, as a dynamic and consistently advancing field, blossoms with accuracy, productivity, and flexibility. One of the persevering difficulties faced by software development organizations is the exact estimation and examination of the time put resources into assorted development assignments. Inconsistencies and inaccuracies in time tracking can have cascading effects, impacting project progress assessment, client billing accuracy, contractor payments, and the refinement of future project estimations. The need for a robust and versatile solution to address these challenges became increasingly evident.

The TMT project emerged against this backdrop of necessity, fueled by the desire to provide software development organizations with a sophisticated yet user-friendly tool to monitor and manage time effectively. The origins of this project stem from the recognition that precise time tracking is not merely an operational task but a strategic endeavor that can empower project managers, enhance client relations, and sharpen the competitive edge in the software development industry.

Purpose of TMT Report:

1. Firstly, to provide a comprehensive understanding of the TMT project:

- Highlighting the timeliness and indispensability of the TMT project.
- Delving into challenges faced by software development companies in time tracking.
- Underscoring the critical need for a dedicated tool like TMT.
- Examining potential pitfalls of inadequate time tracking, including:
 - Project delays
 - Cost overruns
 - Inaccurate invoicing
- Highlighting the ripple effects on an organization's:
 - Bottom line
 - Reputation

2. Secondly, to outline the primary objectives and functions of the TMT system:

- Articulating how TMT empowers developers and project managers by offering:
 - Seamless time recording
 - Precise measurement of time
 - Efficient data consolidation

- Robust security measures
- Elucidating how TMT transforms raw time data into actionable insights.
- Facilitating more informed project planning and development.

Actors listed in the TMT System

Project Manager: The project Manager is in charge of planning, carrying out, and concluding projects, the project manager is a key player in project management. Project Manager: The project Manager is in charge of planning, carrying out, and concluding projects, the project manager is a key player in project management. Project Managers make sure that projects are finished on time, within budget, and scope.

Interpretation: Project Managers are the driving force behind successful project execution. They use the Time Monitoring Tool to transform project plans into actionable steps. This includes setting timelines, allocating resources, and defining milestones. Through the tool, they gain real-time visibility into project progress and can make data-driven decisions to optimize resource allocation, reduce delays, and ensure project success. In essence, Project Managers leverage the Time Monitoring Tool as a strategic ally in their mission to lead projects to fruition.

Developer: Developers are the architects and builders of software applications or systems. They are responsible for coding, testing, and maintaining software components.

Interpretation: Developers are the hands-on creators of the digital solutions that drive projects. Within the Time Monitoring Tool, they use it to meticulously track the time they invest in coding, debugging, and other development tasks. Additionally, they categorize their activities to provide granular insights into their workflow. The tool enables them to collaborate effectively by sharing task updates with Project Managers and team members, fostering transparency and accountability. By analyzing time data, developers can identify areas for efficiency improvement, leading to enhanced productivity and the successful delivery of high-quality software solutions.

Use case list and Use Case Diagram

Use Case List

1. **Recording Timestamps:** The developer Records the Timestamps to log the exact start and end times of the coding sessions and other project-related activities. This ensures that the developer has an accurate record of the time he has invested in each task, aiding in self-assessment and performance improvement.

2. **Activity Categorization:** Activity Categorization is a crucial feature for the Developer. Developers use it to categorize and label their work activities, such as coding, testing, design, or meetings, within the Time Monitoring Tool. This enables developers to provide a detailed breakdown of how they allocate their time, allowing them to track and assess their productivity in specific project areas. It also aids in generating insightful reports that help the team to better understand how their time is utilized across different project tasks.
3. **User Authentication:** User Authentication is crucial for managers as it ensures that only authorized personnel have access to the Time Monitoring Tool. This security feature safeguards sensitive project data and maintains the integrity of our time-tracking system.
4. **Data Analysis:** Data Analysis empowers me to analyze time data collected by the tool. I can identify work patterns, inefficiencies, or areas requiring optimization, allowing me to make informed decisions for resource allocation and project planning.
5. **Resource Tracking:** Resource Tracking provides real-time visibility into the availability and workload of team members and developers. It helps the manager to balance resource allocation, assign tasks effectively, and avoid resource overloading, ultimately enhancing project efficiency.
6. **Weekly Time Sheets:**
 - Developer Perspective: Weekly Time Sheets provide developers with a structured way to compile and report their time allocations throughout the week. This functionality streamlines the reporting process, allowing developers to submit accurate and comprehensive timesheets for review by the Project Manager.
 - Project Manager Perspective: Weekly Time Sheets are an invaluable resource for the Project Manager. They offer an overview of how developers have distributed their time across various project activities during the week. This insight enables the manager to assess progress, identify areas of focus, and maintain transparency in project tracking and reporting.
7. **Task Prioritization:** Task Prioritization enables the manager to set priorities for project tasks, ensuring that critical activities receive immediate attention. This functionality aids in optimizing resource allocation and guiding the team's focus toward high-priority tasks.
8. **Time Budgeting:** Time Budgeting allows managers to allocate predefined time budgets to specific tasks or projects. It aids in effective time management, ensures adherence to project timelines, and helps in controlling project costs by monitoring time spent against budgeted time.

USE CASE DIAGRAM FOR TMT

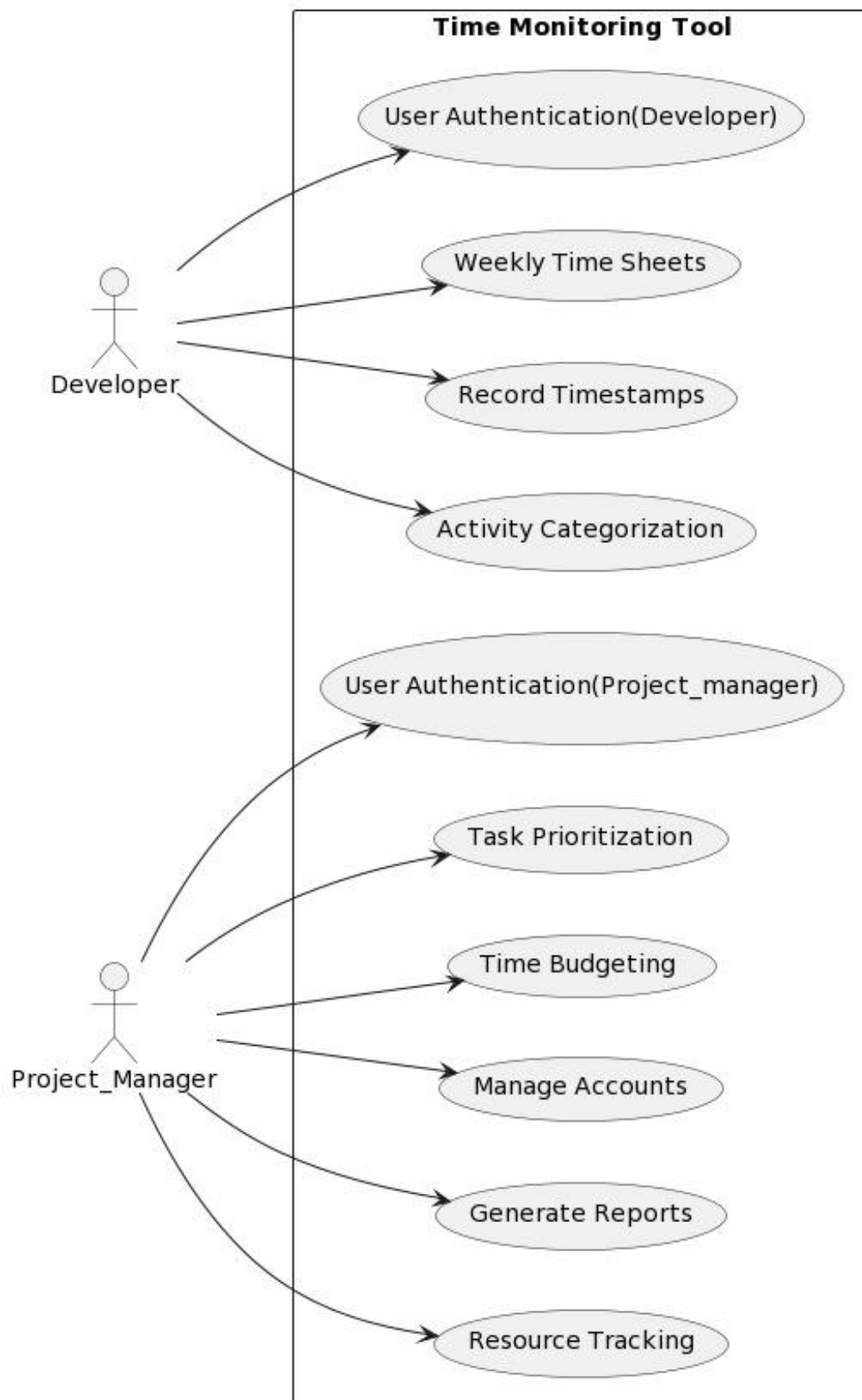


Fig 1: Use Case Diagram for the TMT

Activity Diagrams

1. User Authentication: Activity Diagram for Developer & Project Manager

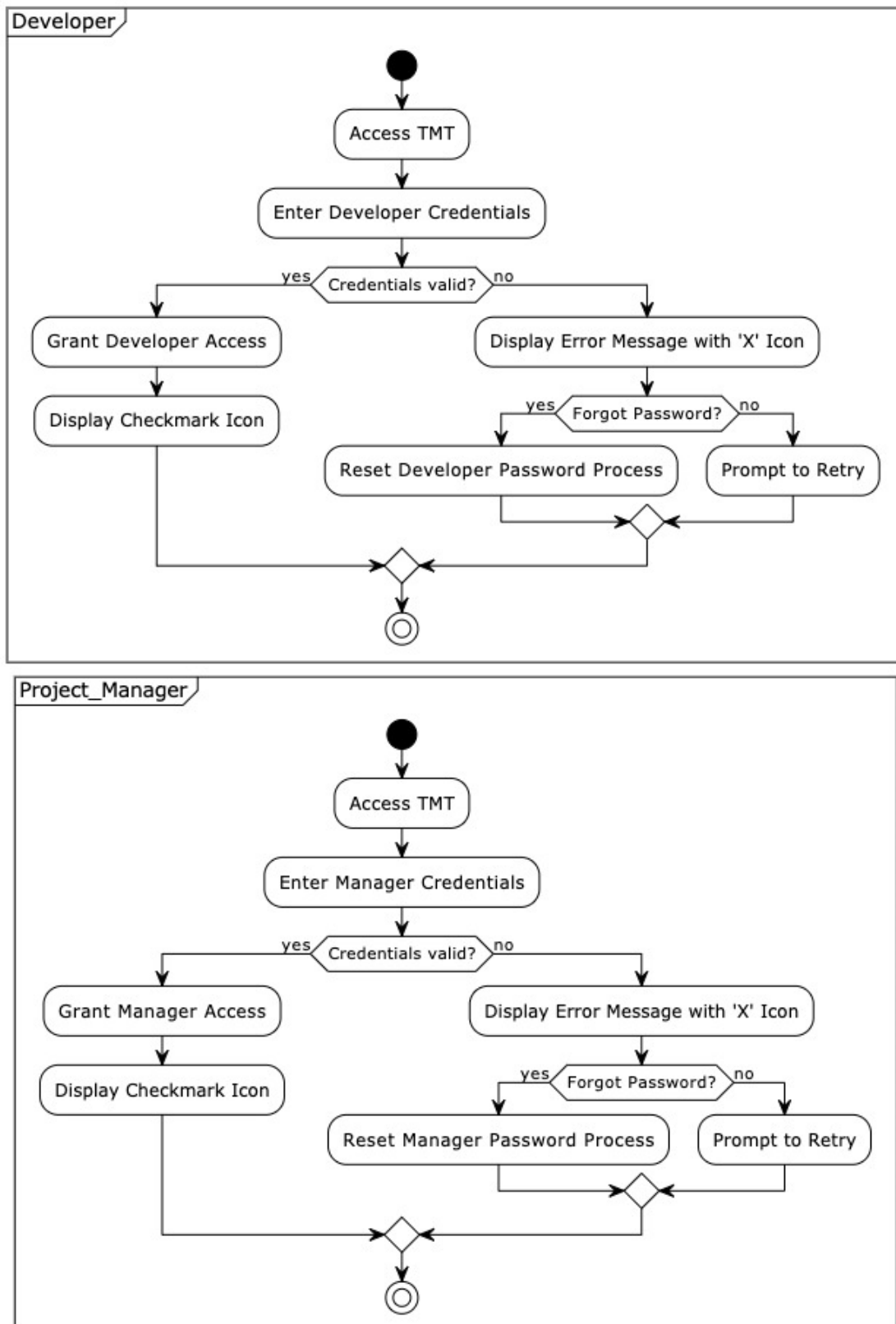


Fig 2.1: Activity Diagram for User Authentication

2. Weekly Time Sheets

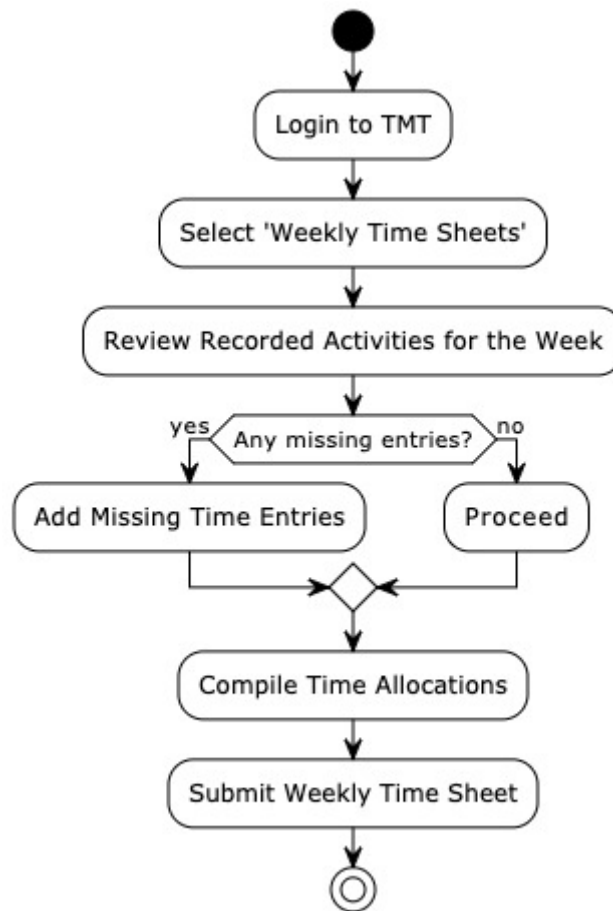


Fig 2.2: Activity Diagram for Weekly Time Sheets

3. Record Timestamps

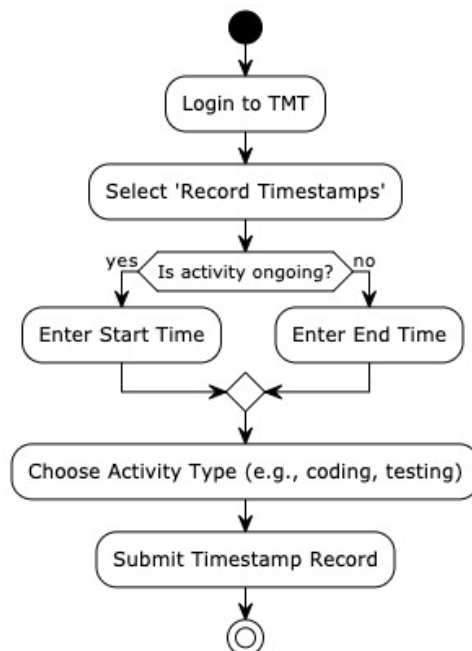


Fig 2.3: Activity Diagram for Recording Timestamp

4. Activity Categorization

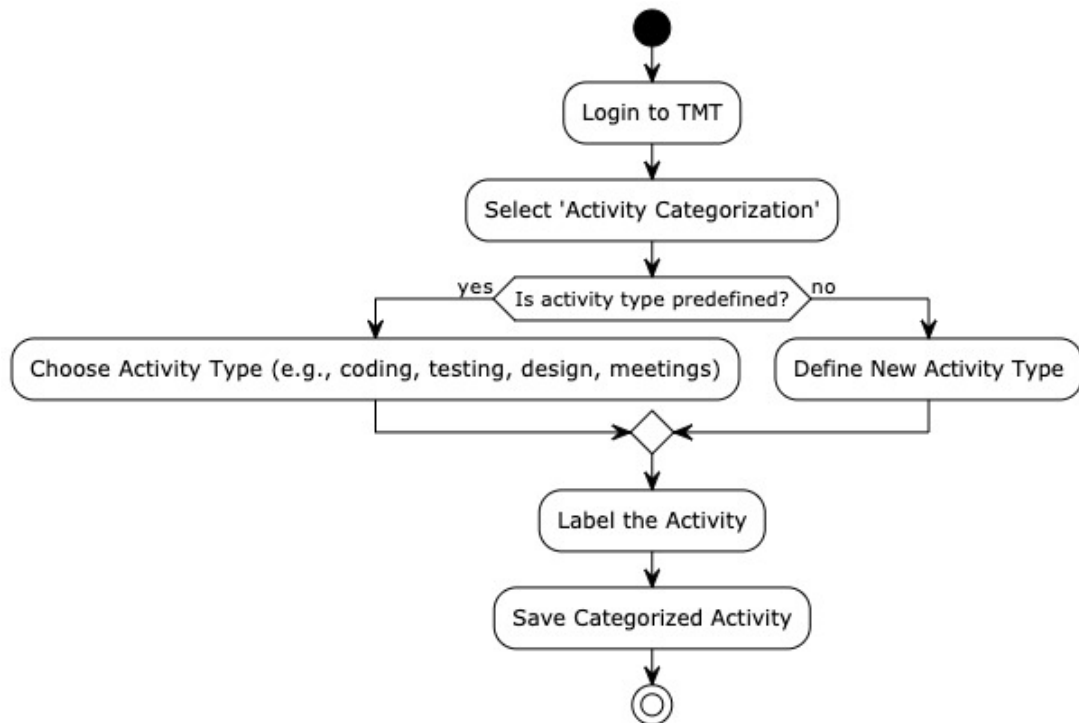


Fig 2.4: Activity Diagram for Activity Categorization

5. Time Budgeting

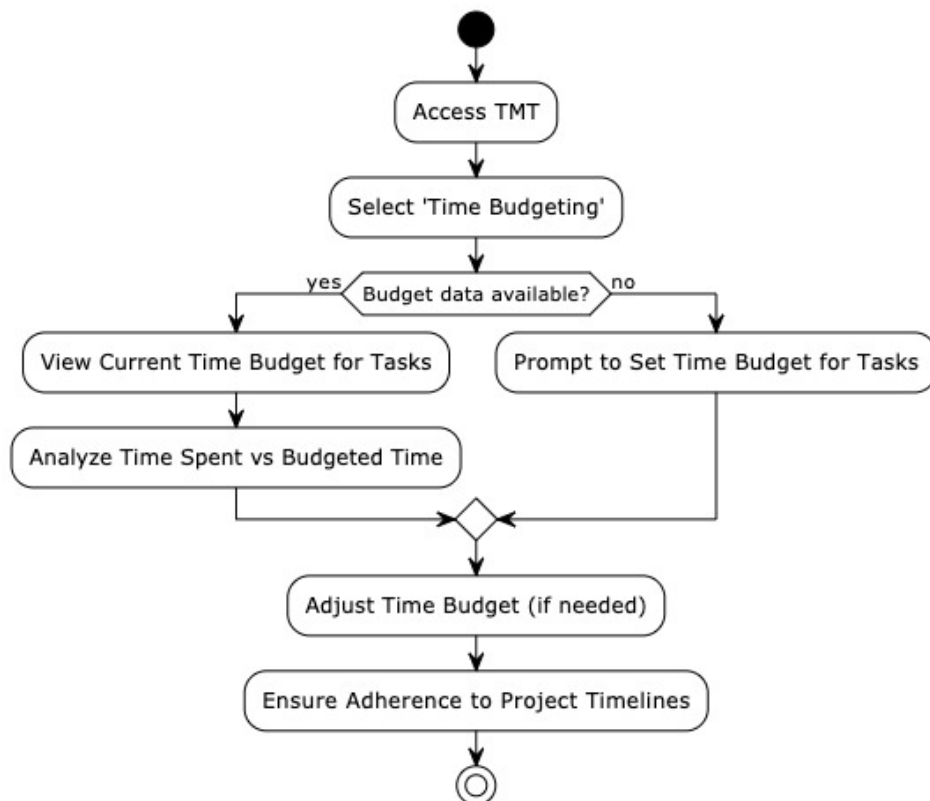


Fig 2.5: Activity Diagram for Time Budgeting

6. Manage Accounts

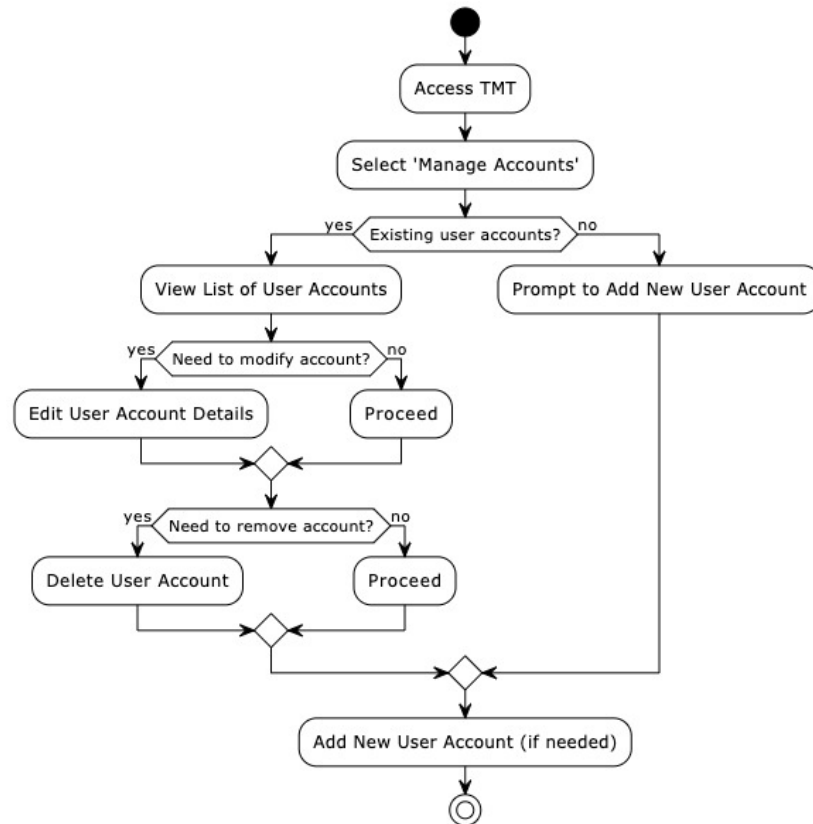


Fig 2.6: Activity Diagram for Manage Accounts

7. Generate Reports

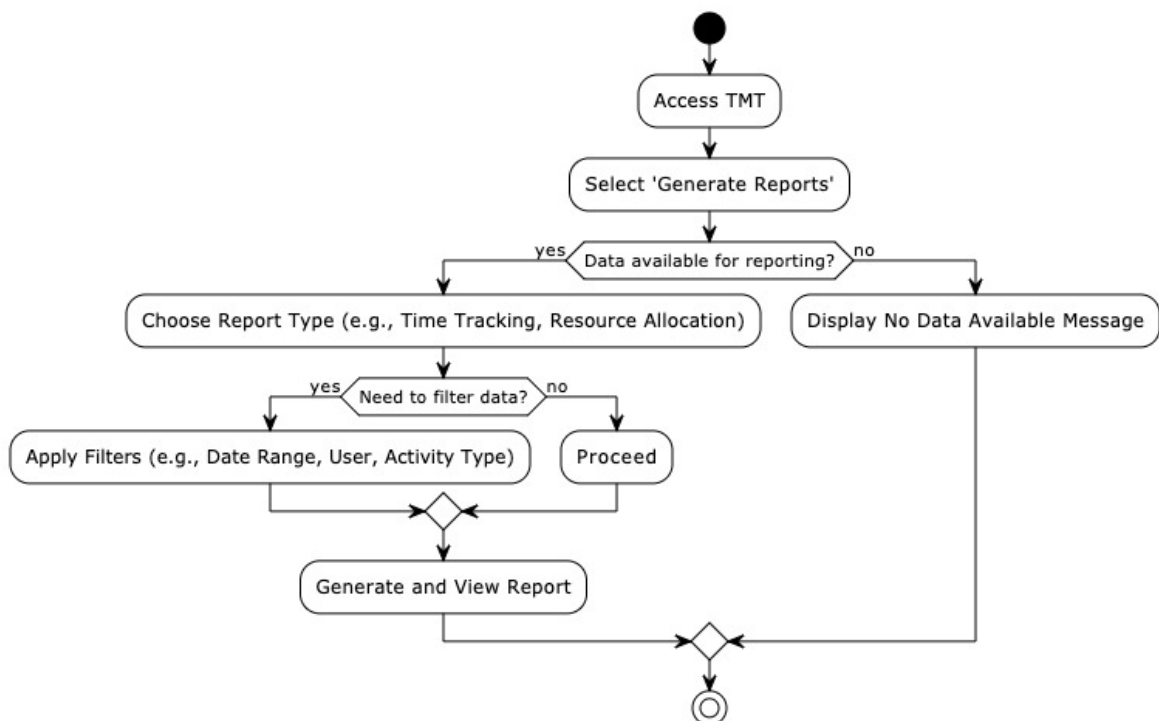


Fig 2.7: Activity Diagram for Generate Reports

8. Resource Tracking

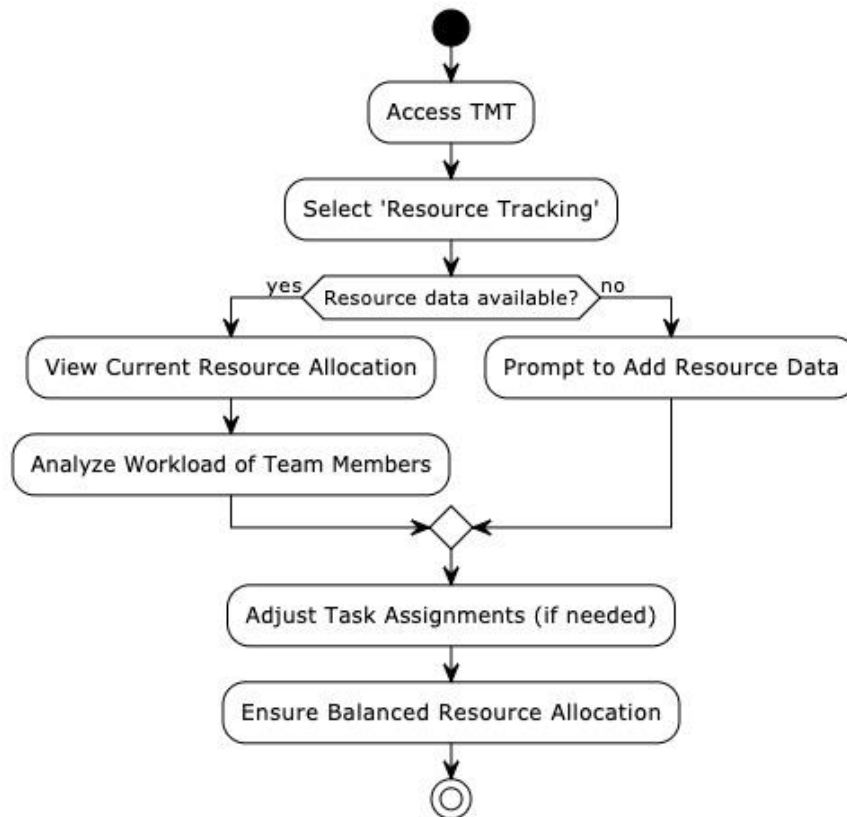


Fig 2.8: Activity Diagram for Resource Tracking

USE CASES

Use Case 1: Recording Timestamps

Actors: Developer

Description: A developer records timestamps to log the exact start and end times of coding sessions and other project-related activities within the Time Monitoring Tool. This enables accurate time tracking for self-assessment and performance improvement.

Use Case 2: Activity Categorization

Actor: Developer

Description: Developers use activity categorization to label and categorize their work activities, such as coding, testing, design, or meetings, within the Time Monitoring Tool. This enables a detailed breakdown of time allocation, facilitating productivity assessment and the generation of insightful reports for project understanding.

Use Case 3: User Authentication

Actors: Developer, Project Manager

Description: User authentication is crucial to ensure that only authorized personnel have access to the Time Monitoring Tool. This security feature safeguards sensitive project data and maintains the integrity of the time-tracking system.

Use Case 4: Data Analysis

Actor: Project Manager

Description: Data analysis empowers project managers to analyze time data collected by the tool. It helps in identifying work patterns, inefficiencies, and areas requiring optimization. This analysis allows for informed decisions regarding resource allocation and project planning.

Use Case 5: Resource Tracking

Actor: Project Manager

Description: Resource tracking provides real-time visibility into the availability and workload of team members and developers. It assists the manager in balancing resource allocation, assigning tasks effectively, and avoiding resource overloading, ultimately enhancing project efficiency.

Use Case 6: Weekly Time Sheets

Actors: Developer, Project Manager

Description:

- **Developer Perspective:** Developers use weekly time sheets to compile and report their time allocations throughout the week in a structured manner. This functionality streamlines the reporting process, allowing for accurate and comprehensive timesheets for review by the project manager.
- **Project Manager Perspective:** Weekly time sheets offer an overview of how developers have distributed their time across various project activities during the week. This insight enables the manager to assess progress, identify areas of focus, and maintain transparency in project tracking and reporting.

Use Case 7: Task Prioritization

Actor: Project Manager

Description: Task prioritization enables the manager to set priorities for project tasks, ensuring that critical activities receive immediate attention. This functionality aids in optimizing resource allocation and guiding the team's focus toward high-priority tasks.

Use Case 8: Time Budgeting

Actor: Project Manager

Description: Time budgeting allows managers to allocate predefined time budgets to specific tasks or projects. It aids in effective time management, ensures adherence to project timelines, and helps in controlling project costs by monitoring time spent against budgeted time.

These use case descriptions provide a detailed understanding of each use case's purpose, the actors involved, and the functionality it offers within the Time Monitoring Tool (TMT) system.

Verification & Validation

To verify and validate the business processes and functional models for the Time Monitoring Tool project, let's review each component as follows:

Actors List:

- The actor list includes two primary actors: the Project Manager and the Developer, which is consistent with the project's context.
- The interpretations of the roles of these actors are well-defined and align with their responsibilities within the system.

Use Case Diagram:

- The use case diagram accurately represents the interactions between actors and use cases in the system.
- All essential use cases, as described in the use case descriptions, are included in the diagram.
- The relationships (associations, extends, includes) appear to be appropriately depicted based on the descriptions.

Activity Diagrams:

- Activity diagrams for each use case have been mentioned but are not provided in the text. Activity diagrams are crucial for visualizing the workflow within each use case.
- To complete the verification and validation process, it's essential to create and validate activity diagrams for each use case, ensuring they accurately represent the workflow and steps involved.

Use Case Descriptions:

- The use case descriptions are well-structured and provide detailed overviews of each use case.
- Each description includes essential information, such as the use case's name, ID number, type, primary actor, brief description, importance level, stakeholders, and triggers.
- The flow of events, including normal flow, sub-flows, and alternate flows, is documented clearly and coherently.
- There is consistency between the descriptions and the corresponding activity diagrams, although activity diagrams are not provided in the text.