



**Dhirubhai Ambani**  
**University**  
Technology

Formerly DA-IICT

## IT 314 - Software Engineering Project Report Group 24

September 24, 2025

### Group Members

|                             |           |
|-----------------------------|-----------|
| Raiyani Rudra Chetanbhai    | 202301223 |
| Patel Apurv Ashokbhai       | 202301230 |
| Ajudiya Kashyap Jagdishbhai | 202301239 |
| Shreyas Dutta               | 202301246 |
| Rishik Yalamanchili         | 202301258 |
| Chirag Katkoriya            | 202301259 |
| Mahek Jikkar                | 202301260 |
| Patel Nakul Jaymitkumar     | 202301261 |
| Jalu Rishabh Devdanbhai     | 202301265 |
| Siddhant Shekhar            | 202301268 |



GitHub Repository Link

# Contents

|           |  |           |
|-----------|--|-----------|
| <b>1</b>  | <b>Introduction</b>                          | <b>3</b>  |
| <b>2</b>  | <b>Concept Poster and Chart Paper</b>        | <b>3</b>  |
| <b>3</b>  | <b>Identify Stakeholders &amp; Users</b>     | <b>4</b>  |
| <b>4</b>  | <b>Elicitation Techniques</b>                | <b>5</b>  |
| 4.1       | End-Users (Freelancers, Employees) . . . . . | 5         |
| 4.2       | Managers . . . . .                           | 5         |
| 4.3       | Developer Team . . . . .                     | 5         |
| <b>5</b>  | <b>Identify the FRs and NRFs</b>             | <b>6</b>  |
| 5.1       | Functional Requirements (FR) . . . . .       | 6         |
| 5.2       | Non-Functional Requirements (NFR) . . . . .  | 7         |
| <b>6</b>  | <b>User Stories</b>                          | <b>8</b>  |
| 6.1       | General User Stories . . . . .               | 8         |
| 6.2       | User Stories for End-Users . . . . .         | 11        |
| 6.3       | User Stories for Project Managers . . . . .  | 12        |
| <b>7</b>  | <b>EPICs</b>                                 | <b>13</b> |
| <b>8</b>  | <b>POC for SPRINT 1 (Implementation)</b>     | <b>15</b> |
| <b>9</b>  | <b>GitHub project repository link</b>        | <b>15</b> |
| <b>10</b> | <b>Contributions</b>                         | <b>15</b> |

# 1 Introduction

This project outlines the development of an intelligent, cross-platform time tracking system designed to eliminate the burden of manual logging. The system automatically captures user activity across devices, which an AI classifies against specific projects and tasks. This automated data collection provides features, including insightful graphical reports, AI-generated work summaries, invoice generation. The platform is built to improve individual productivity while providing managers with an accurate real-time view of team efforts.

## 2 Concept Poster and Chart Paper

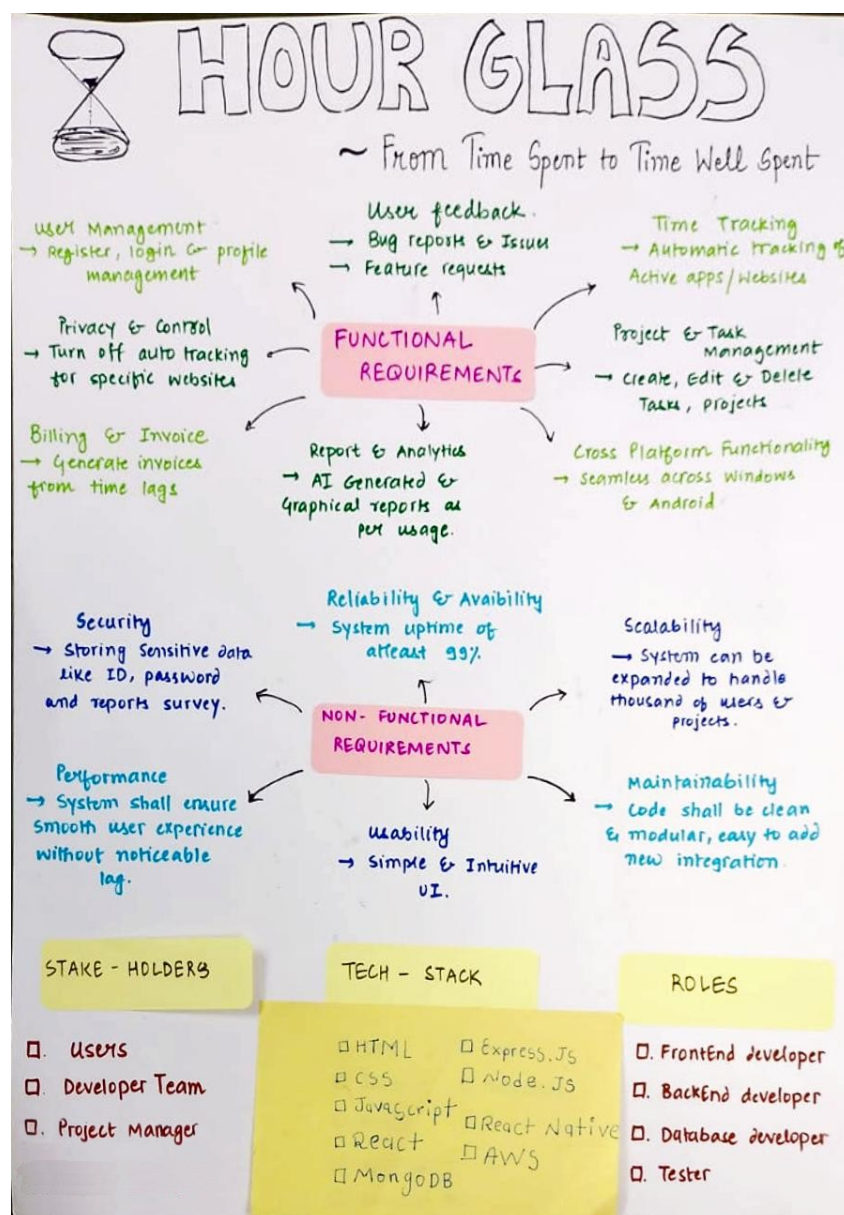


Figure 1: Concept Poster

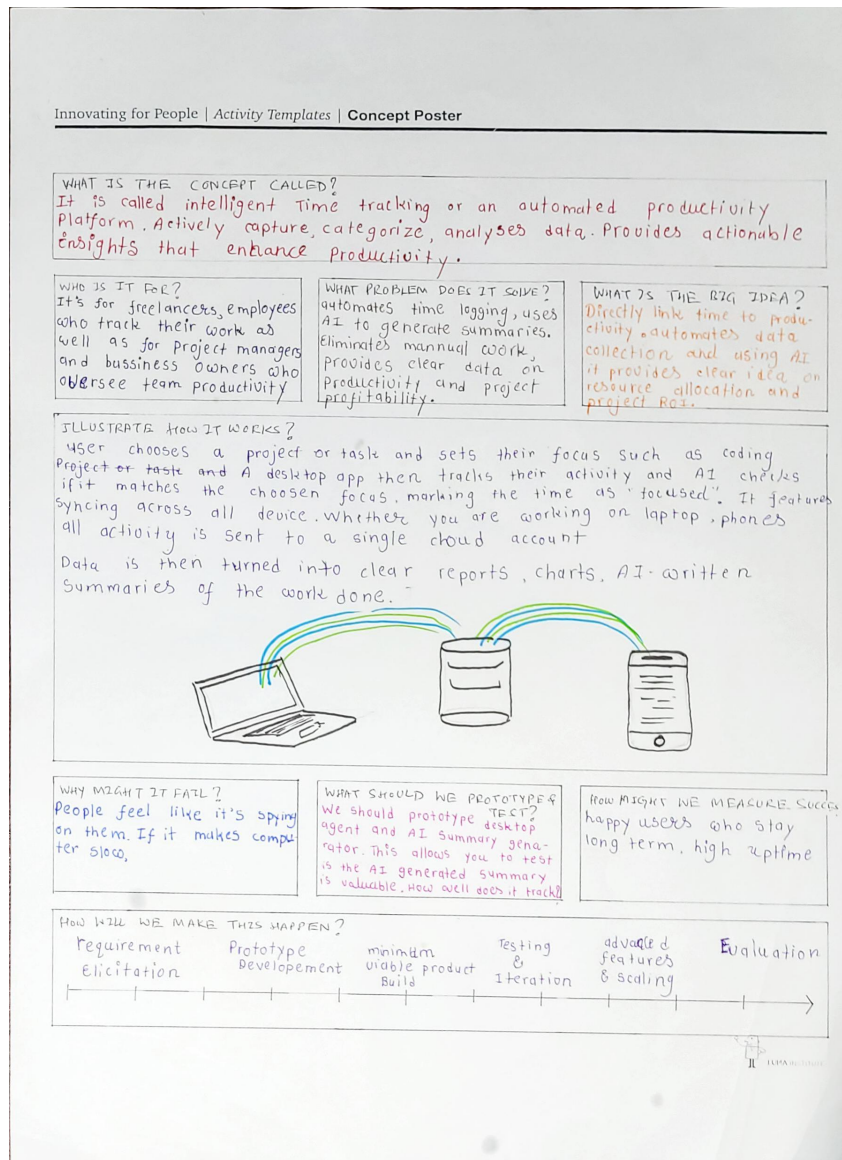


Figure 2: Chart Paper

### 3 Identify Stakeholders & Users

- **Freelancers**  
users who track billable and non- billable hours across multiple device.
- **developer Team**  
team invloved in the Development of this App.
- **Project Manager**  
Oversees progress, assigns tasks, and ensures deadlines and quality are met.

## 4 Elicitation Techniques

### 4.1 End-Users (Freelancers, Employees)

This group will use the app daily. Their primary concern is that the app is useful, easy to use, and respects their privacy

#### **Technique 1: Surveys & Questionnaires**

*Justification:* Efficient way to gather quantitative and qualitative data from a large number of users. Helps identify desired features, current problem points in time tracking.

#### **Technique 2: Prototyping & User Feedback Sessions**

*Justification:* Since the app emphasizes a “Simple & Intuitive UI”, observing user interactions validates design choices before coding. Directly addresses the need to test the “desktop agent and the AI Summary Generator”.

#### **Technique 3: Interviews**

*Justification:* One-on-one conversations reveal deeper insights that surveys miss. Open-ended questions such as “Walk me through how you currently track your work time” help uncover workflows and opportunities for improvement.

### 4.2 Managers

This group focuses on team productivity, project profitability, and data-driven decision making.

#### **Technique 1: Structured Interviews**

*Justification:* Managers have specific business goals. A structured interview with prepared questions ensures that all critical points are covered, such as: “What metrics do you need in team reports?” or “How do you generate invoices from tracked time?” This defines features like Reporting & Analytics and Billing & Invoicing.

### 4.3 Developer Team

This group is concerned with technical feasibility, architecture, and long-term maintenance of the system.

#### **Technique 1: Brainstorming & Technical Workshops**

*Justification:* Internal sessions allow collaboration on architecture, tech stack (Java, React, PostgreSQL, etc.) and feasibility of features such as AI summaries. Ensures maintainability and scalability are planned from the start.

#### **Technique 2: Analysis of Existing Systems**

*Justification:* Studying competitor apps reveals standard features, usability practices, and market gaps. This avoids pitfalls and provides insights for performance, design, and functionality.

## 5 Identify the FRs and NRFs

### 5.1 Functional Requirements (FR)

- **FR-1: Time & Attendance Tracking**

- FR-1.1: The system shall provide an automated time tracker that records activity based on application usage.
- FR-1.2: Users shall be able to classify time entries as “Billable” or “Non-Billable.”
- FR-1.3: The system shall detect periods of user inactivity (idle time) and allow the user to discard that time from a running timer.
- FR-1.4: The system shall provide a workflow for employees to submit weekly/bi-weekly timesheets.
- FR-1.5: The system shall allow managers to review, approve, and reject submitted timesheets.
- FR-1.6: The system shall provide a “Remember Me” option on the login screen. If selected, the system shall store a secure authentication token that keeps the user logged in for 15 consecutive days without requiring re-entry of credentials.

- **FR-2: Project & Team Management**

- FR-2.1: Users shall be able to create projects and tasks.
- FR-2.2: Administrators shall be able to create user accounts and assign roles (e.g., Admin, Manager, Member).
- FR-2.3: Managers shall be able to assign projects and tasks to specific team members.

- **FR-3: Productivity & Monitoring**

- FR-3.1: The system shall maintain an audit log of changes made to time entries.
- FR-3.2: The system shall generate productivity metrics based on keyboard and mouse activity levels. (Conflicts)

- **FR-4: Finance & Billing**

- FR-4.1: The system shall generate professional invoices from billable hours tracked.
- FR-4.2: The system shall be able to calculate payroll data based on hours worked and predefined pay rates. (Conflict)

- **FR-5: Reporting & Analytics**

- FR-5.1: The system shall generate standard reports (e.g., time by project, time by user, time by date range).
- FR-5.2: Users shall be able to create, save, and export reports.

- FR-5.3: The system shall provide visual dashboards summarizing key metrics.
- FR-5.4: The system shall allow for scheduling reports to be automatically generated and emailed.

## 5.2 Non-Functional Requirements (NFR)

- **NFR-1: Usability**

- NFR-1.1: The user interface (UI) shall be intuitive and require minimal training for a new user to start tracking time.
- NFR-1.2: Core actions, such as starting and stopping a timer, must be accessible from the main dashboard within a single click.
- NFR-1.3: The system shall provide clear visual feedback for all user actions.

- **NFR-2: Performance**

- NFR-2.1: The web application pages shall load in under 1 second on a standard internet connection.
- NFR-2.2: API responses for critical functions shall be returned in under 500ms.
- NFR-2.3: The system must support a minimum of a few thousand concurrent users without significant performance degradation.

- **NFR-3: Compatibility & Accessibility**

- NFR-3.1: The system shall be accessible via modern web browsers (Chrome, Firefox, Safari, Edge).
- NFR-3.2: The system shall offer dedicated desktop applications for Windows, macOS, and Linux.
- NFR-3.3: The system shall offer dedicated mobile applications for iOS and Android.
- NFR-3.4: The desktop and mobile applications must support offline time tracking, with data synced automatically once a connection is restored.

- **NFR-4: Security**

- NFR-4.1: All user data, especially passwords and payment information, must be encrypted in transit and at rest.
- NFR-4.2: The system shall provide role-based access control to ensure users can only access data relevant to their permissions.

- **NFR-5: Reliability & Availability**

- NFR-5.1: The system shall have a service uptime of at least 99.9%.
- NFR-5.2: The system must ensure data integrity, with no loss of time-tracking data once it is recorded.
- NFR-5.3: The system must have a regular backup and disaster recovery plan in place.

- **NFR-6: Interoperability & Extensibility**

- NFR-6.1: The system shall provide a well-documented public API for third-party developers.
- NFR-6.2: The system shall offer native integrations with popular project management (e.g., Jira) and accounting software.

## 6 User Stories

### 6.1 General User Stories

#### User Story 1: Sign Up

**As** a new user (end user, freelancer, employee, manager, or owner),  
**I want** to sign up and create my account,  
**So that** I can access the platform securely.

**Success Criteria:**

- User enters valid details (email/phone, password, role if required).
- System creates the account and sends confirmation (email/OTP).
- User receives a success message and can log in.

**Failure Criteria:**

- Email/phone already exists → “Account already exists.”
- Password doesn’t meet strength requirements → error shown.
- OTP/email verification fails → prevent activation.

#### User Story 2: Login

**As** a registered user,  
**I want** to log in using my credentials,  
**So that** I can securely access my account.

**Success Criteria:**

- Correct email/username and password entered.
- System authenticates and redirects to dashboard.
- Login session is created securely.

**Failure Criteria:**

- Invalid credentials → “Incorrect username or password.”
- Account inactive/not verified → “Please verify your account.”

#### User Story 3: Remember Me Option

**As** a registered user,  
**I want** a “Remember Me” option,  
**So that** I can stay logged in for 15 days.

**Success Criteria:**

- Checkbox “Remember Me” displayed at login.



- Secure token stored for 15 days.
- Session ends after logout or expiry.

**Failure Criteria:**

- Auto-logout not triggered after 15 days.
- Manual logout does not end session.

#### **User Story 4: Password Reset**

**As** a user who forgot their password,  
**I want** to reset it securely,  
**So that** I can regain access.

**Success Criteria:**

- User clicks “Forgot Password.”
- System sends reset link/OTP.
- New strong password set successfully.
- System confirms reset and login works.

**Failure Criteria:**

- Email/phone not registered → “Account not found.”
- OTP/link expired/invalid → error shown.

#### **User Story 5: Role-Based Access Control**

**As** a system user,  
**I want** role-based access,  
**So that** sensitive data is protected.

**Acceptance Criteria:**

- Freelancers can view/edit only their entries.
- Managers can manage only their projects.
- Clients see only their invoices/reports.
- Admins have full access.
- Unauthorized → “Permission denied.”

#### **User Story 6: Data Privacy & Encryption**

**As** a system user,  
**I want** encryption in storage and transit,  
**So that** my data remains secure.

**Acceptance Criteria:**

- Sensitive data is encrypted in the database.
- All traffic uses HTTPS.
- Insecure connection → “Secure connection required.”

## User Story 7: Archive a Project

As a user,

I want to archive projects,

So that my active list stays clean.

### Success Criteria:

- Project archived and hidden from active list.
- Archived data still available in reports.

### Failure Criteria:

- Project not found.
- Already archived.
- Unauthorized action.
- System error during archiving.

## User Story 8: AI-Powered Categorization

As a user,

I want AI categorization of tasks/projects,

So that I save time on classification.

### Success Criteria:

- AI suggests categories.
- User can override suggestions.
- System learns from corrections.

### Failure Criteria:

- AI fails → “Uncategorized.”
- Service unavailable → “AI categorization failed.”

## User Story 9: Mobile Accessibility

As an employee,

I want mobile access,

So that I can log hours remotely.

### Success Criteria:

- Mobile-responsive UI.
- Offline logging supported.

### Failure Criteria:

- Unsupported device/browser → error shown.

## User Story 10: Export Reports

As a manager,

I want export options,

So that I can share reports.

### Acceptance Criteria:

- Export as PDF, CSV, XLSX.
- Errors → “Failed to export report.”

## 6.2 User Stories for End-Users

### Automatic Time Capture

As an employee,  
I want auto start/stop timers,  
So that I don't forget to log hours.

**Success Criteria:**

- Timer starts with system activity.

**Failure Criteria:**

- Timer fails → "Start manually."

### View Time Report

As an employee,  
I want weekly reports,  
So that I can track productivity.

**Success Criteria:**

- System displays hours/project breakdowns.

**Failure Criteria:**

- Invalid week.
- Access denied.
- Report generation error.

### Apply for Leave

As an employee,  
I want leave requests,  
So that time is not tracked during absence.

**Success Criteria:**

- Leave dates selected, system confirms.

**Failure Criteria:**

- Invalid/overlapping dates.
- Permission denied.
- Network error.

### Offline Time Tracking

As a freelancer,  
I want offline tracking,  
So that work is tracked without internet.

**Success Criteria:**

- Data stored locally and synced later.

**Failure Criteria:**

- Storage full.
- Sync conflict or failure.
- Unauthorized offline use.

## 6.3 User Stories for Project Managers

### Assign Tasks

As a project manager,  
**I want** to assign tasks,  
**So that** work is distributed efficiently.

**Acceptance Criteria:**

- Valid user assignment with deadline.
- Errors if user not found, deadline invalid, or already assigned.

### Quick Meeting Setup

As a project manager,  
**I want** a "Set Meet" button next to an employee's name,  
**So that** can instantly schedule and log a non-billable meeting.

**Success Criteria:**

- A non-billable time entry is automatically created for employee.

### Manage Expenses

As a project manager,  
**I want** expense approval,  
**So that** costs remain within budget.

**Success Criteria:**

- Budget defined, expenses logged.
- Manager approves/rejects.

**Failure Criteria:**

- Invalid budget or missing rates.
- Budget exceeded.
- Permission denied.

### Monitor Real-Time Progress

As a manager,  
**I want** dashboards,  
**So that** I can monitor milestones.

**Success Criteria:**

- Dashboard updates in real time.

**Failure Criteria:**

- No projects assigned.
- Timeout/loading error.
- Permission denied.
- Data inconsistency.

### **Set Delay Alerts**

**As** a project manager,  
**I want** alerts for delayed tasks,  
**So that** I can act proactively.

#### **Acceptance Criteria:**

- Alerts trigger on missed deadlines.
- Errors if unassigned, duplicate, or invalid.

### **Approve Invoice**

**As** a manager,  
**I want** invoice approval,  
**So that** billing aligns with deliverables.

#### **Acceptance Criteria:**

- Valid invoice approved.
- Errors for missing, duplicate, or unverified entries.

### **Provide Feedback**

**As** a Manager,  
**I want** to give feedback to employees,  
**So that** collaboration improves.

#### **Success Criteria:**

- Feedback stored and notified.

#### **Failure Criteria:**

- Empty form, duplicate, invalid rating, or network error.

## **7 EPICs**

### **Epic 1: User Authentication & Account Management**

This covers all functionality related to user identity, access, and security.

- Sign Up
- Login
- Remember Me Option
- Password Reset

### **Epic 2: Core Time Tracking**

This covers the fundamental features for an individual to log and manage their own time.

- Automatic Time Capture
- Offline Time Tracking
- Apply for Leave

### **Epic 3: Project & Task Management**

This covers the organizational structure of how work is defined and assigned.

- Archive a Project
- Assign a Task to a Team Member

### **Epic 4: Team Collaboration & Management**

This covers features specifically for managers to oversee their teams.

- Monitor Real-Time Progress
- Set a Delay Alert

### **Epic 5: Reporting & Analytics**

This covers all features related to viewing, analyzing, and exporting time data.

- View Time Report
- Export Reports

### **Epic 6: Billing & Financials**

This covers the entire workflow from tracking expenses to invoicing and getting client feedback.

- Manage Project Expenses Within Budget
- Generate an Invoice
- Approve an Invoice
- Provide Feedback on Service

### **Epic 7: Intelligent Platform Features**

This is a dedicated epic for the advanced AI and automation capabilities.

- AI-Powered Categorization of Tasks & Projects

## Epic 8: System-Wide Requirements

This covers concerns that apply to the entire application.

- Role-Based Access Control
- Data Privacy & Encryption
- Mobile Accessibility

## 8 POC for SPRINT 1 (Implementation)

Here is a link to our POC video.

## 9 GitHub project repository link



GitHub Repository Link

## 10 Contributions

| Member                      | Contribution   |
|-----------------------------|--|
| Chirag Katkoriya            | Brainstorming, Frontend, User Stories  |
| Ajudiya Kashyap Jagdishbhai | Brainstorming, Analysis of Existing Time Tracking Systems, Functional Requirements |
| Patel Nakul Jaymitkumar     | Brainstorming, Frontend, Identifying Stakeholders                                  |
| Mahek Jikkar                | Backend, User Stories, Brainstorming   |
| Raiyani Rudra Chetanbhai    | Backend, Brainstorming, Analysis of Existing Time Tracking Systems                 |
| Shreyas Dutta               | Brainstorming, Backend, Interviews   |
| Patel Apurv Ashokbhai       | Brainstorming, Backend, User Stories   |
| Siddhant Shekhar            | Brainstorming, Frontend, Non-Functional Requirements                               |
| Jalu Rishabh Devdanbhai     | Brainstorming, Interviews  |
| Rishik Yalamanchili         | Brainstorming, Survey  |