

Document 6- Please prepare a use case diagram, activity diagram and a use case specification document

Use Case Name	Description	Actors	Basic Flow	Alternate Flow	Exceptional Flows
Automated Reminders	System sends reminders to employees for pending timesheet submissions.	Primary: System Scheduler Secondary: Employee	Identify pending → Fetch emails → Send reminder	Reminder sent via in-app notification	Email server failure → retry or log error
Escalation Notifications	Notify reporting manager when employee fails to submit timesheet after reminders.	Primary: System Secondary: Manager	Check reminder status → Identify escalations → Notify manager	Escalation sent to alternate contact	Incorrect manager mapping → log for admin correction
Manager Dashboard	Manager views timesheet status of team in real-time.	Primary: Manager	Login → Access dashboard → View submission stats	Filter by date/team/member	Dashboard not loading → show maintenance message
Role-Based Access	Users can only access features assigned to their roles.	Primary: Admin Secondary: All Users	Login → Verify role → Grant relevant access	Temporary elevated access by Admin	Unauthorized access → redirect or deny access
Timesheet Data Export	Managers/HR can export timesheet data in preferred format.	Primary: Manager, HR	Select period → Choose format → Generate & download	Auto-schedule reports for weekly download	Export service down → retry or alert admin
Audit Log Generation	System logs key actions like submissions, approvals, and escalations.	Primary: System Secondary: Auditor	Capture events → Store logs securely → View logs	Export logs for compliance	Log service failure → queue and retry
Integration with HRMS	Sync employee data and hierarchy from HRMS system.	Primary: System Secondary: HR Admin	Connect to HRMS → Fetch data → Sync records	Scheduled batch sync at night	HRMS API error → skip record and log failure

Predictive Compliance Engine	Predict potential non-compliant users based on history.	Primary: System Secondary: Compliance Officer	Load historical data → Run prediction → Display risk scores	Weekly prediction updates	Data model failure → fallback to manual reports
-------------------------------------	---	--	---	---------------------------	---

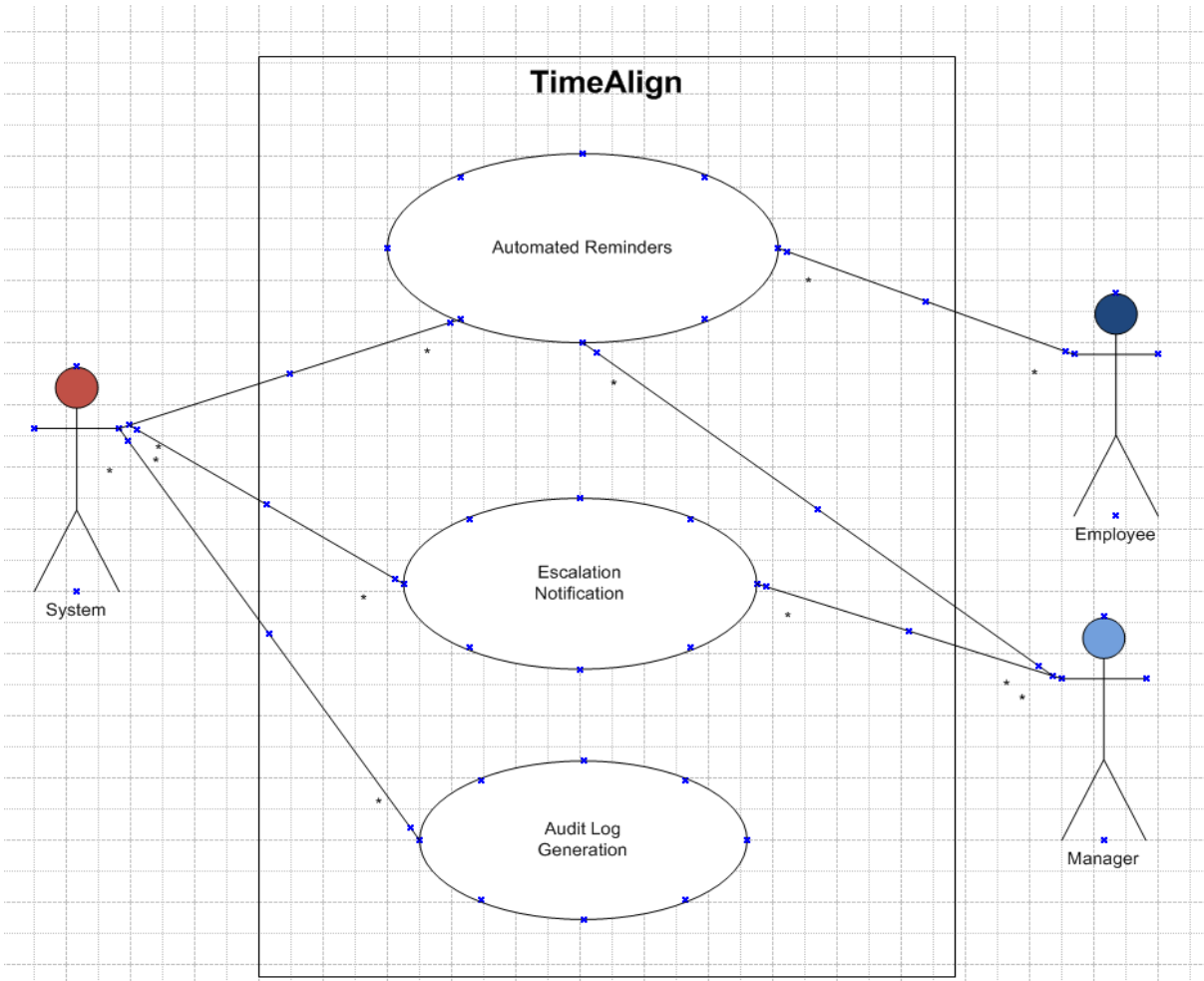
Use Case Name	Pre-Conditions	Post-Conditions	Assumptions	Constraints	Dependencies
Automated Reminders	Timesheet period open; user records exist	Reminders sent and logged	Users have valid email addresses	Email must follow format and be sent during business hours	Mail server, employee directory
Escalation Notifications	Reminder window expired; timesheet still pending	Escalation sent to manager	Correct reporting structure is configured	Only after specific grace period	Reporting hierarchy, notification system
Manager Dashboard	Manager logged in with access rights	Dashboard data displayed	Manager role mapped correctly	Real-time updates may be limited to batch refresh	Database, role management service
Role-Based Access	User authenticated and role assigned	Access limited as per role definition	Role definitions are accurate and updated	Roles cannot be changed dynamically during session	User directory, access control service
Timesheet Data Export	Valid export request made by authorized user	Data downloaded in selected format	Data is complete and available	Export file format must match allowed types (CSV, XLS)	Reporting module, timesheet DB
Audit Log Generation	User/system action performed	Entry logged securely	Logging system is active	Log cannot be altered or deleted by end users	Logging service, secured storage
Integration with HRMS	HRMS system accessible; API keys active	Employee data updated	HRMS contains accurate and up-to-date employee records	Sync can only occur during maintenance window	HRMS API, network, employee master

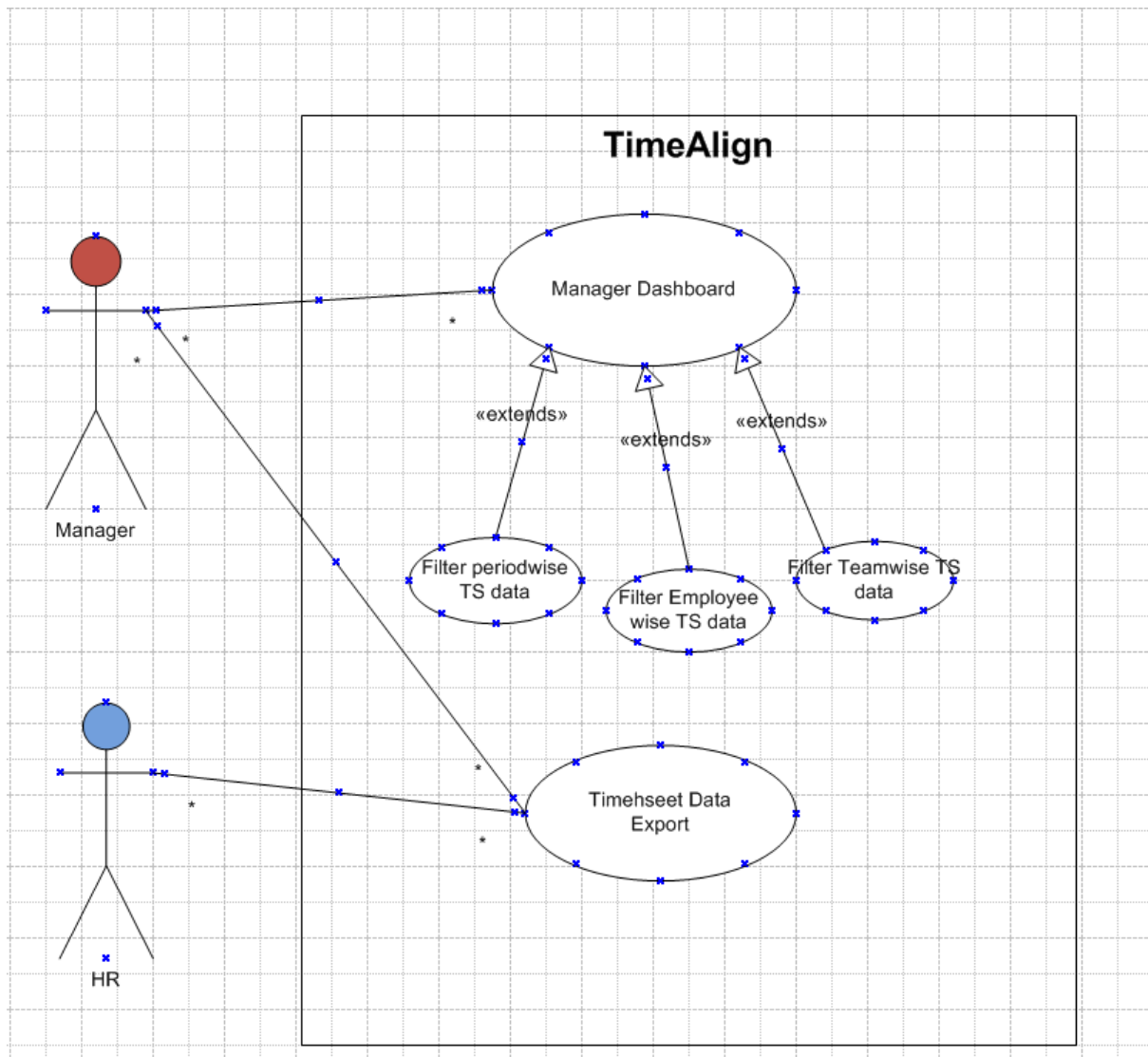
Predictive Compliance Engine	HRMS API, network, employee master	Risk score generated	Sufficient past data for meaningful prediction	Model retraining interval fixed; interpretability required	ML model, analytics service, timesheet data
-------------------------------------	------------------------------------	----------------------	--	--	---

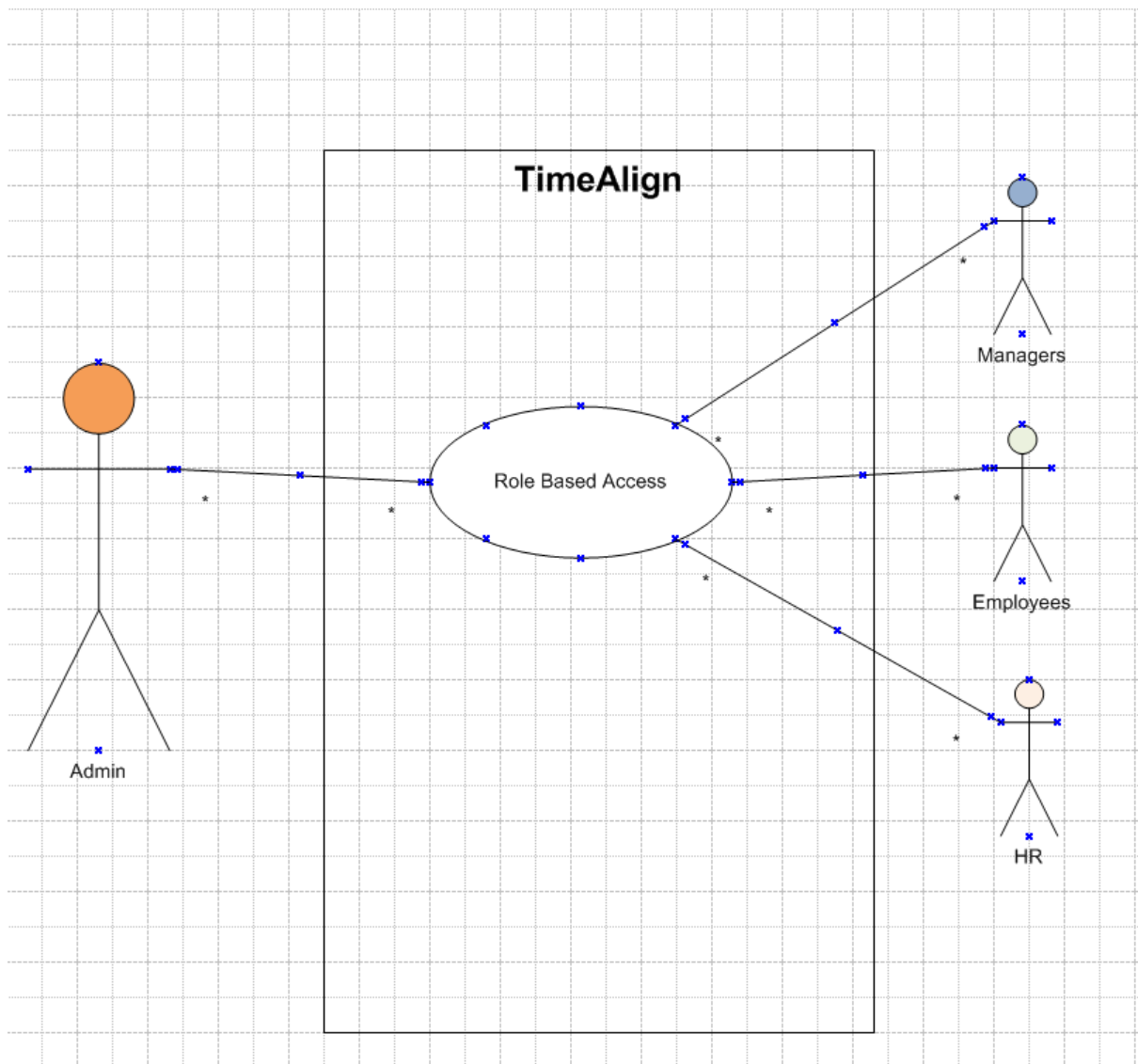
Use Case Name	Inputs & Outputs	Business Role	Misc. Info
Automated Reminders	Input: User deadlines Output: Reminder notifications	- Reminders must trigger 24 hours before due date - Only active employees receive reminders	Scheduled via CRON or batch job
Escalation Notifications	Input: Missed deadlines Output: Alert sent to reporting manager	- Escalation only if user hasn't submitted timesheet post 48 hrs of reminder	Based on user's reporting hierarchy from HRMS
Manager Dashboard	Input: Team timesheet data Output: Summary and status visuals	- Only managers with mapped team IDs can access - Data shown must be real-time or near real-time	May include charts, filters, date-range search
Role-Based Access	Input: User login credentials Output: Access to permitted features only	- Each role (Employee, Manager, Admin) has defined access levels	Controlled through an access control matrix
Timesheet Data Export	Input: Date range, format selection Output: Downloadable file	- Only authorized roles can export - File must be encrypted if includes sensitive data	Supports CSV, XLS; includes header, timestamp
Audit Log Generation	Input: User/system actions Output: Logged entry with timestamp	- Logs are immutable - Logs must be retained for 5 years	Stored securely with access via admin role only
Integration with HRMS	Input: Employee master data	- Sync only allowed during maintenance window	API-based integration with token authentication

	Output: Synced user profile info	- No duplicate employee IDs allowed	
Predictive Compliance Engine	Input: Past timesheet behavior Output: Risk score or alert recommendation	- Manual overrides must be logged - Scores must be explainable and compliant with internal policy	Powered by ML model; periodic model retraining planned

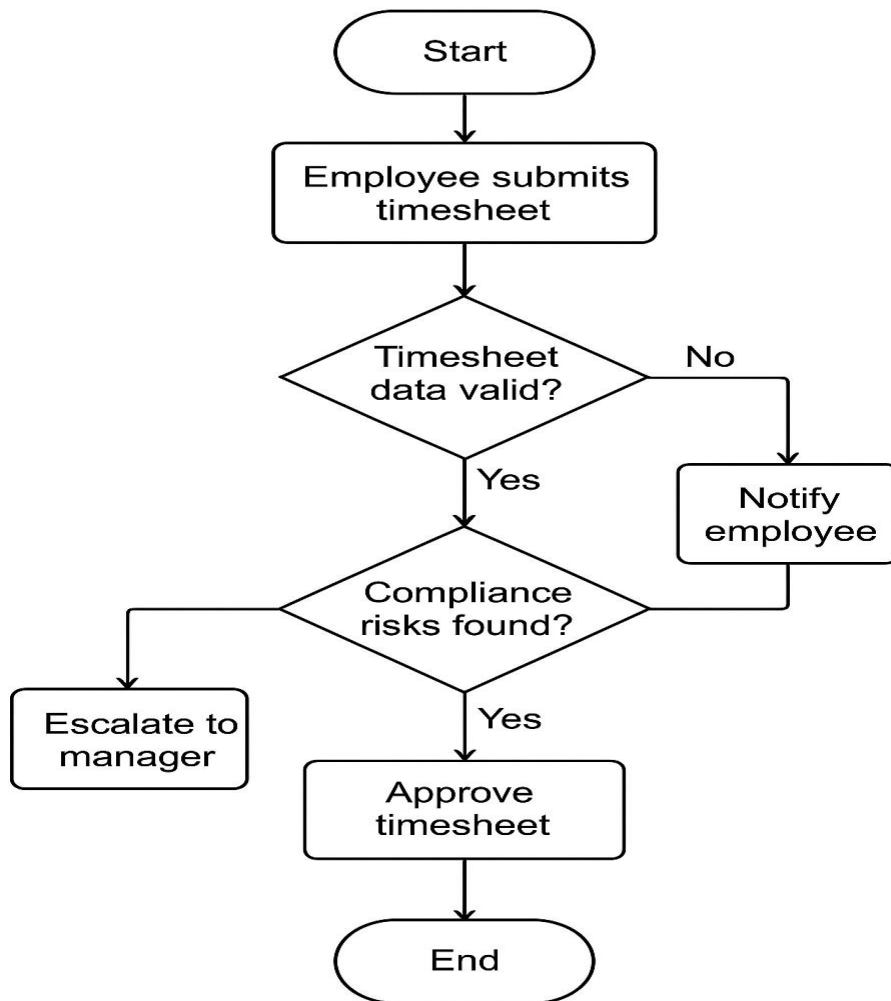
USE CASES





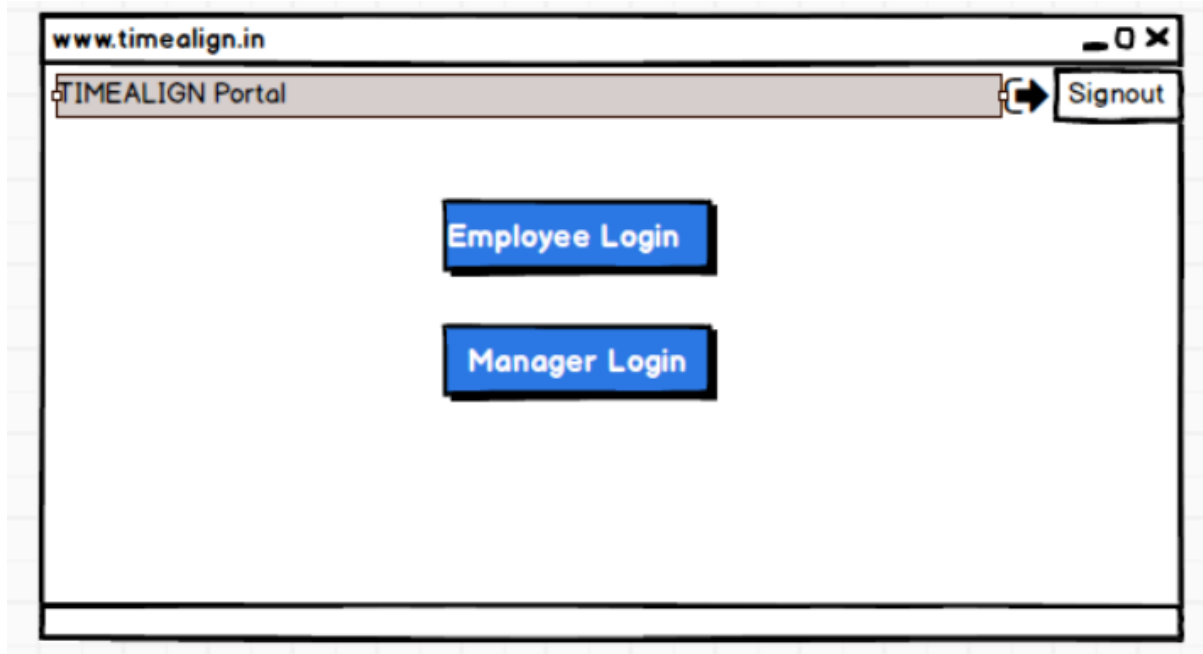


Activity Diagram

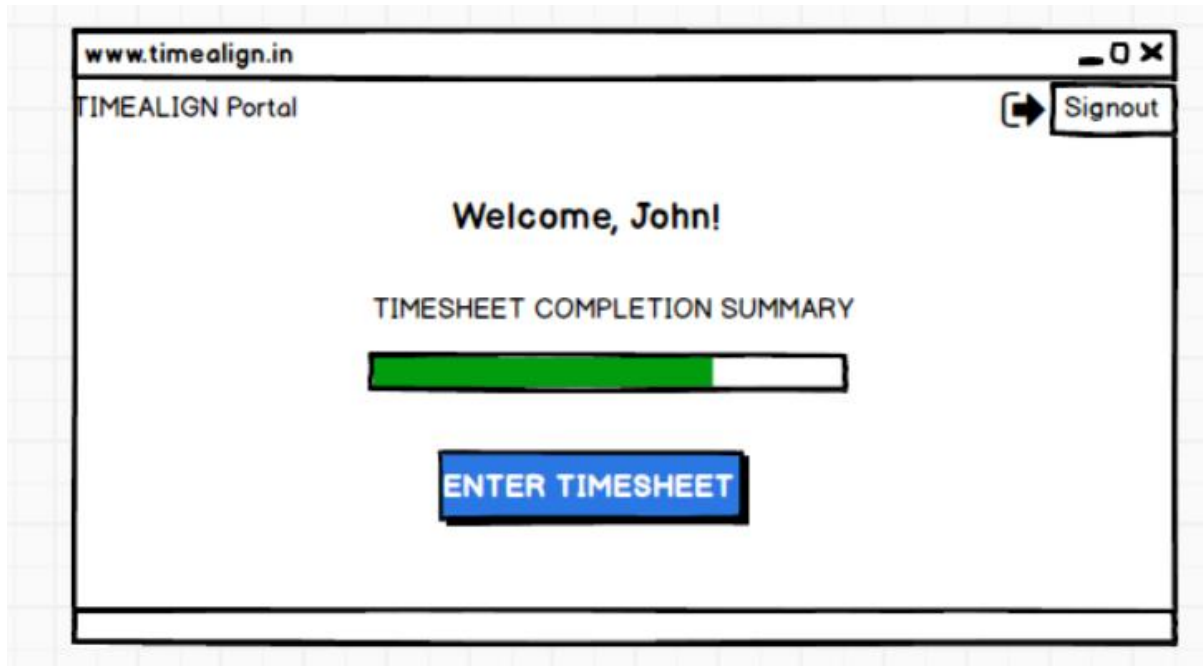


Document 7 : Screen And Pages

1) Home Screen



2) Employee Login Screen



3)Manager Login Screen

The screenshot shows the 'Manager Login Screen' of the TIMEALIGN Portal. The browser address bar displays 'www.timealign.in'. The page header includes 'TIMEALIGN Portal' and a 'Signout' button. The main content area features a 'Welcome, John!' message, a blue 'Dashboard' button, and a 'TIMESHEET COMPLETION SUMMARY FOR TEAM' section with a green progress bar. Below this is a table titled 'Defaulting Employees'.

Employee	Status
1)Jane	Not Submitted
2)Smith	Not Submitted
3) Emily	Not Approved

4)Dashboard Screen

The screenshot shows the 'Dashboard Screen' of the TIMEALIGN Portal. The browser address bar displays 'www.timealign.in'. The page header includes 'TIMEALIGN Portal' and a 'Signout' button. The main content area contains three sections, each with a dropdown menu and a list of options:

- Track Timesheet Report for current Month**
 - teamwise
 - datewise
- Previous Timesheet Reports**
 - Teamwise
 - Periodwise
- Practicewise Timesheet Report**
 - DCH
 - COA1

5) Export Functionality

The screenshot shows a web application interface for 'www.timealign.in'. The main heading is 'Timesheet Report'. Below it, there is a dropdown menu labeled 'Export Report' with options 'Excel' and 'PDF'. A 'Signout' button is located in the top right corner. Below the export options, there is a section titled 'Timesheet Report For June 2025'. This section contains a table with the following data:

Name (job title)	Employee id	TS Filled	TS Pending for Approval
Giacomo Guilizzoni Founder & CEO	40	45%	20%
Marco Botton Tuttofare	38	98%	80%
Mariah MacLachlan Better Half	41	60%	NIL
Valerie Liberty Head Chef	51	20%	5%
Data Grid Docs			☐

Document 8- Tools-Visio and Axure Write a paragraph on your experience using Visio and Axure for the project.

For this project, I primarily used **Microsoft Visio** and **Balsamiq** to support documentation and visualization of business requirements. **Visio** was instrumental in designing professional and clear **use case diagrams, activity diagrams**, and other UML artifacts, which helped communicate the system's behaviour to stakeholders and the development team. Additionally, I used **Balsamiq** to create low-fidelity wireframes of key application screens. Its simple drag-and-drop interface made it easy to design mock-ups quickly and focus on layout and functionality without getting into visual details. Together, these tools enhanced collaboration across teams and ensured that both functional and non-functional requirements were well-understood during the Waterfall project lifecycle.

Document 9 – Business Analyst Experience.

As a Business Analyst, I was actively involved across key phases of the project lifecycle, particularly in requirement gathering and requirement analysis. Below is a summary of my experience:

1. Requirement gathering :

- I adopted the MoSCoW technique to categorize and prioritize requirements into Must Have, Should Have, Could Have, and Won't Have. This helped manage stakeholder expectations effectively.
- During this phase, the client was intermittently unavailable, so I proactively identified alternative points of contact from their end to avoid project delays and collected the necessary inputs in a timely manner.
- I applied the FURPS model (Functionality, Usability, Reliability, Performance, and Supportability) to validate and evaluate the completeness and quality of gathered requirements.
- I identified and eliminated several redundant and duplicate requirements, ensuring clarity and avoiding confusion during later phases.
- I used prototyping techniques to help stakeholders better visualize the requirements, which encouraged more precise and practical inputs.

2. Requirement Analysis:

- I created various UML diagrams, including use case diagrams and activity diagrams, to clearly illustrate the functional requirements and process flows.
- These diagrams were shared with the cross-functional team for discussion. As a BA, I handled feedback from team members, facilitated discussions, and incorporated necessary modifications based on collaborative inputs.
- I was responsible for preparing detailed Business Requirement Specification (BRS) and Software Requirement Specification (SRS) documents, aligning them with both business needs and technical feasibility.

3. Design:

- Derived **test cases** directly from use case diagrams to ensure traceability.
- Engaged with the client to finalize **design and solution documents**.
- Authored both **positive and negative test cases**, ensuring coverage of all potential scenarios.

- Emphasized the importance of **not missing test cases**, as omissions can affect downstream development.
- Created relevant **test data** and ensured the **RTM (Requirement Traceability Matrix)** was continuously updated to reflect design decisions.

4. Development:

- Facilitated JAD (Joint Application Development) sessions to align development goals.
- Acted as the bridge between the technical team and business, clarifying requirements during implementation.
- Addressed conflicts or lack of cooperation by engaging in one-on-one discussions, promoting a healthy team environment.
- Encouraged developers to refer to visual diagrams (UML, activity flows) while coding.
- Coordinated regular meetings between the client and development team, handled attendance challenges by recording sessions and conducting follow-ups.

5. Testing:

- Prepared test cases using earlier use case documentation as a base.
- Participated in high-level functional testing to validate core business flows.
- Coordinated with the client to obtain realistic test data.
- Kept the RTM up-to-date and ensured complete coverage.
- Obtained client sign-off post testing and supported User Acceptance Testing (UAT) readiness.

6. Deployment:

- Sent the finalized RTM to the client, attaching it to the project closure document.
- Coordinated the creation and distribution of end-user manuals.
- Organized and facilitated training sessions for business users and stakeholders.
- Ensured full participation in training, tracked attendance, and addressed gaps proactively.

