**Technical Design Document**

**for**

**Tools Tracker Application**

**Reviewer & Approval History:**

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Signature** |
| **Author** | Nipun Dewangan |  |
| **Reviewed and Approved By** | Sekh Umar |  |

All product names if referenced herein are trademarks of their respective companies.

**Document Management Information**

**Document Title :**  Tools Tracker Application Document

**Document Status :** Final

**Document Publication History:**

(All revisions made to this document must be listed in chronological order)

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **R****emark** |
| 1.0 |  |  | Final |

**Version History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Page** | **Date** | **Description of Change** |
| 1.0 |  |  |  |

Table of Contents

[1. Overview and objectives: 4](#_Toc448135753)

[2. Process Flow: 4](#_Toc448135754)-5

[3. Assumptions:](#_Toc448135755) 6

[4. Functional requirements: 6](#_Toc448135756)-7

[5. Data requirements: 7](#_Toc448135757)

[6. System Architecture: 7](#_Toc448135758)-8

[7. Context Diagram: 9](#_Toc448135759)

[8. Data Flow Diagram: 10](#_Toc448135760)

[9. Logical Data Model: 11](#_Toc448135761)

[10. Physical data structures: 11](#_Toc448135762)-13

[11. Batch Processes: 13-14](#_Toc448135763)

[12. User Interfaces: 14-19](#_Toc448135764)

[13. Appendices 20-22](#_Toc448135765)

### Overview and objectives:

**Tools - which is a commodity for the sustenance of Hirakud Power & Smelter, is sourced from receiving from General Stores through proper channel.**

**Different vendors are supplying materials as per the Purchase Orders placed by the Hirakud Purchase Department, based on the day-to-day maintenance requirements requested by the Maintenance Department.**

### Process Flow:

Tools are stored in the Tools Maintenance Department as requested by the Maintenance Department. TMP prepares the requisition based on the requirements from various maintenance departments and obtains approval from the Department Head, in accordance with the hierarchy and DAO (Delegation of Authority) approval limits. If the requested amount exceeds the defined threshold, approval is sought from the Unit Head.

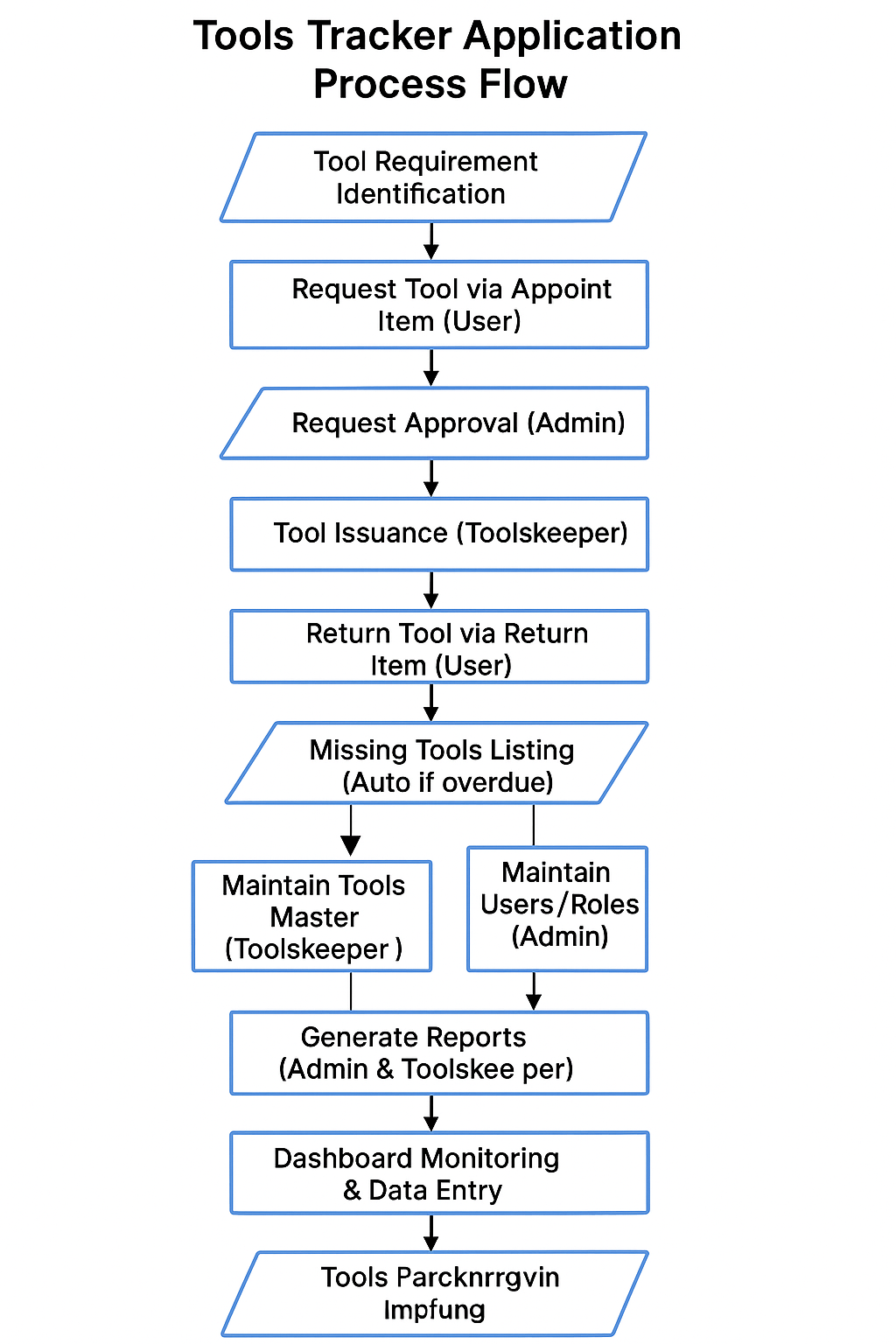
After the requisition is approved, the Purchase Department obtains quotations from different vendors and finalizes the vendor selection based on cost-effectiveness and availability. Once the purchase terms and conditions are agreed upon, the Purchase Department places the order with the selected vendor.

As per the agreed terms, the vendor delivers the materials to the Store Department. The Store Department receives the materials and, with the assistance of the User Department, conducts an inspection. If the materials pass inspection, they are delivered to the Tools Maintenance Department. The Stores Department then raises the MRR (Material Receipt Report) and forwards it along with all necessary documents to the Accounts Department for payment processing.

The Tools Maintenance Department records all received tools in the inventory system, which reflects the current stock levels. Based on approved requests from the Maintenance Department, the Tools Room Keeper issues the tools to the requester along with a return date commitment.

Upon job completion, the user returns the tools to the Tools Room Keeper by the agreed return date.

* 1. **Process Flow Diagram**

****

### Assumptions:

**The software would be designed with the following assumptions:**

1. **Different sets of users from different departments would interact with the system and each set of user would have a defined role. For example, the maintenance user raise the requestion for tools requirement and approver approved the request as per requirements and the tools keeper as per availability of stock issuing the Tools.**

1. **After approval of requesting the tools keeper is issuing the Tools as per FIFO method. (i.e. sequentially check as per approval order)**
2. **The promise date of return must define.**
3. **The Tools keeper distribution the tools as fairly**
4. **The alert is raise, if return due date is over**

### Functional requirements:

* 1. **Functional requirements for maintenance of master data**
     1. **FRMM-0001:**

**The Tools Keeper recorded the Tools name along with update the newly received quantity from General stores.**

* + 1. **FRMM-0002:**

**The Tools keeper create and maintain the user id and name along with others details as required for application.(Manager name, Department, status**

* 1. **Functional requirements for maintenance of transaction data:**
     1. **FRTD-0001:**

**The users must be created the requestion tools quantity for as per his requirements**

* + 1. **FRTD-0002:**

**The Approver must be approved/ reject the requestion as per maintenance needs.**

* + 1. **FRTD-0003:**

**The Admin/Tools keeper as per approved list issuing the tools to concern users as per return promise date**

* + 1. **FRTD-0004:**

**Users are must be return the tools as per promise date**

* 1. **Functional requirements for System Security:**
     1. **FRSS-0001:**

**Users must be provided the facility to interact with the system using one or more defined roles.**

* + 1. **FRSS-0002:**

**Users must be provided the facility to change their respective logon passwords that would be masked while the users key in the same.**

* 1. **Functional requirements for Reporting:**

**The users will have the choice of viewing the output on the screen**

* + 1. **FRR-0001:**

**Users must be provided a report that would display all master information in a single screen.**

* + 1. **FRR-0002:**

**Users must be provided with a report which tools current stock**

* + 1. **FRR-0003:**

**Users must be provided with a report which due date over the promise date**

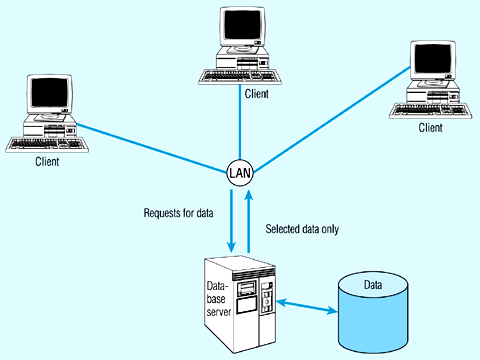
### Data requirements:

**As explained in the previous section, tools item listed and tools receiving qty update as received from General stores, along with current status of tools item quantity update as per transaction.**

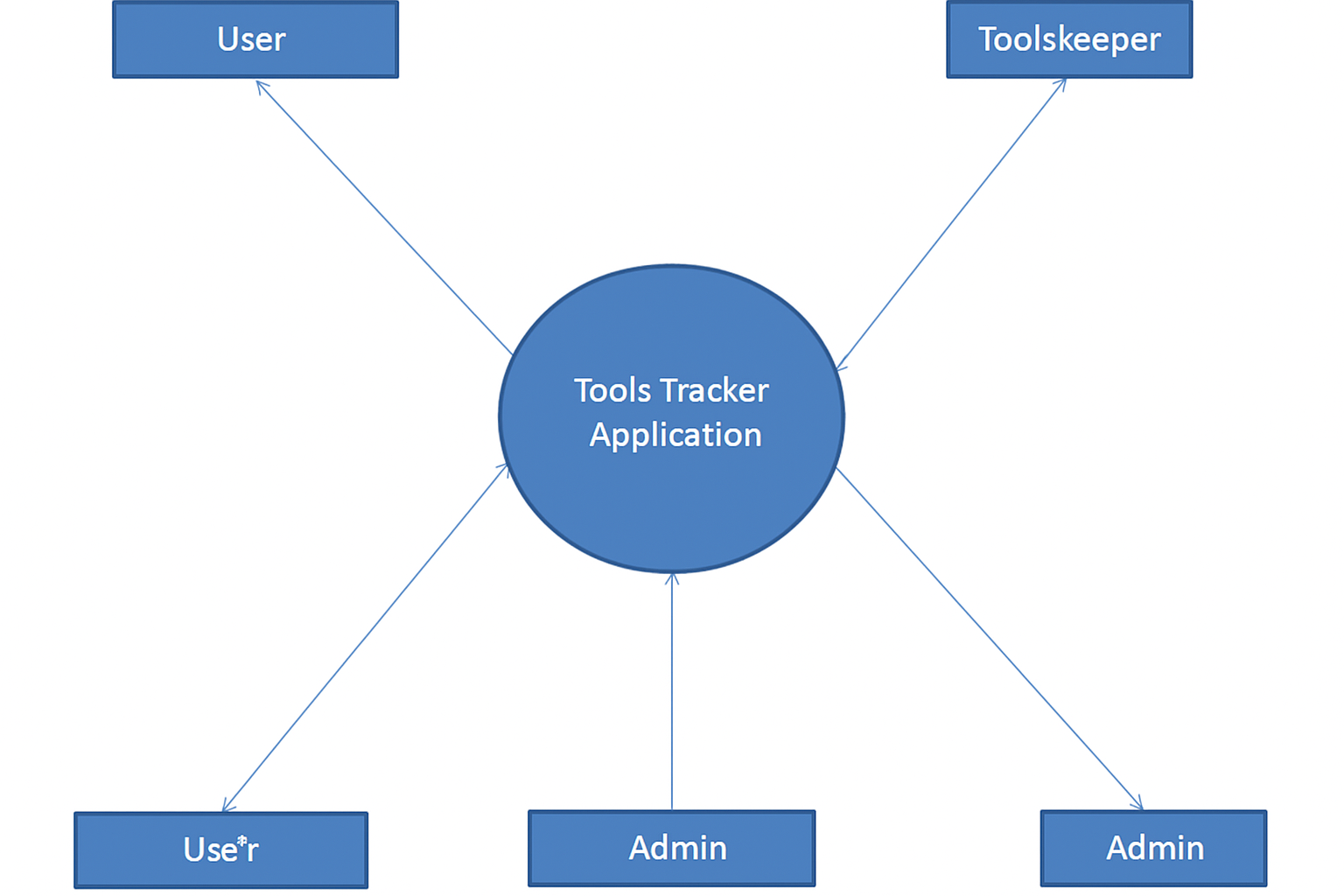
### System Architecture:

**The Tools Tracker application is developed using a modern two-tier client-server architecture leveraging MERN technologies:**

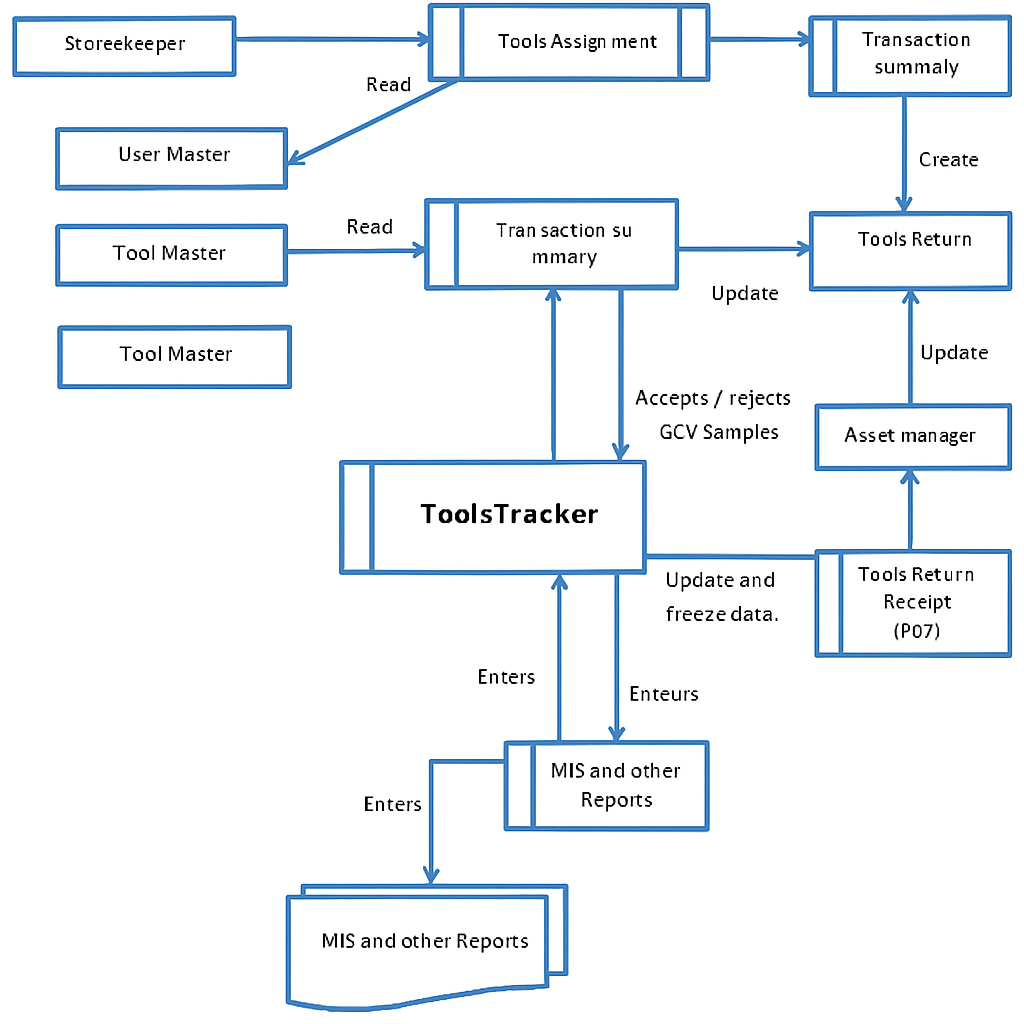
1. **Frontend (Client-side)**
   * **Built with React.js, providing dynamic, responsive, and component-based UI.**
   * **Users (Admin, Toolskeeper, Maintenance Users) interact with the application via a web browser, accessing feature-rich interfaces with React Router for routing and Tailwind CSS for styling.**
   * **The frontend communicates with backend APIs via Axios HTTP calls ensuring seamless data exchange and interactive user experiences.**
   * **Reports are displayed on view-only pages with export capabilities to CSV or Excel formats using JavaScript libraries like SheetJS (xlsx) for Excel integration, replacing Oracle Forms OLE APIs.**
2. **Backend (Application server)**
   * **Developed using Node.js with Express.js, acting as the application server exposing RESTful APIs for all business functionalities such as appointing tools, returning tools, approving requests, user authentication, and role-based authorisation.**
   * **All batch processing logic, business rules, and data validations are implemented as modular controllers and services within Express.js routes, replacing Oracle stored packages for logic encapsulation.**
3. **Database (Data server)**
   * **Uses MongoDB as the back-end NoSQL database for persisting users, tools inventory, requests, appointments, returns, approvals, and logs.**
   * **MongoDB collections replace the traditional relational database tables while ensuring flexible schema management suitable for growing data structures and new features.**

****

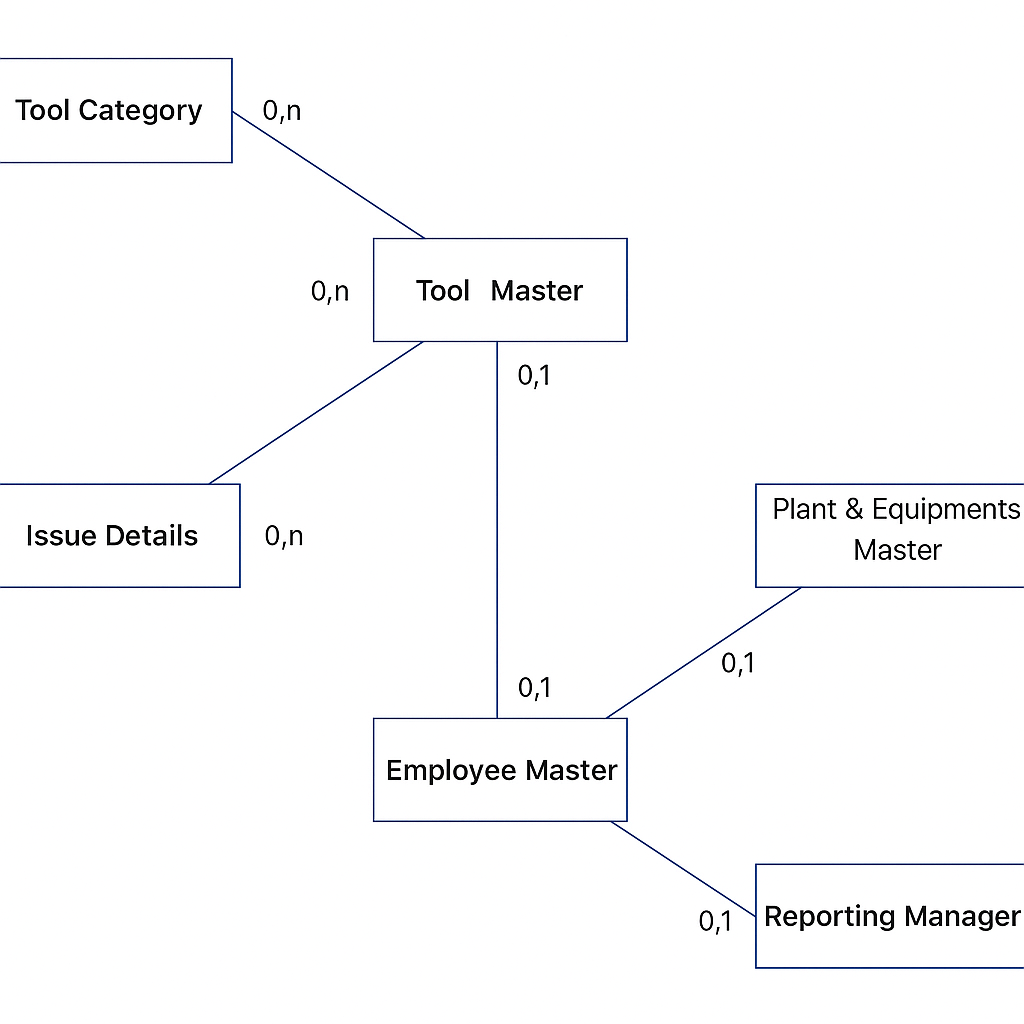
### Context Diagram:



### Data Flow Diagram:

****

### Logical Data Model:

****

### Physical data structures:

**1. Table: tools\_master**

* **Primary Key: tool\_id**
* **Foreign Key: Nil**
* **Description: Stores each tool's name, description, category, quantity, status, purchase details.**

**2. Table: users\_master**

* **Primary Key: user\_id**
* **Foreign Key: Nil**
* **Description: Stores user credentials, role (Admin, Toolskeeper, Maintenance User), email, hashed\_password.**

**3. Table: requests**

* **Primary Key: request\_id**
* **Foreign Key: fk03\_user (user\_id on users\_master)  
  fk01\_tool (tool\_id on tools\_master)**
* **Description: Stores tool requests with status (pending, approved, rejected), timestamps.**

**4. Table: appointed\_tools**

* **Primary Key: appointment\_id**
* **Foreign Key: fk03\_user (user\_id on users\_master)  
  fk01\_tool (tool\_id on tools\_master)**
* **Description: Stores tools appointed to users with appointed date, condition, quantity.**

**5. Table: returned\_tools**

* **Primary Key: return\_id**
* **Foreign Key: fk04\_appointed (appointment\_id on appointed\_tools)**
* **Description: Stores tool return records with return date, condition, remarks.**

**6. Table: tool\_approval\_history**

* **Primary Key: approval\_id**
* **Foreign Key: fk03\_user (user\_id on users\_master)  
  fk02\_request (request\_id on requests)**
* **Description: Stores approval actions by Admin with timestamps.**

**7. Table: missing\_tools**

* **Primary Key: missing\_id**
* **Foreign Key: fk01\_tool (tool\_id on tools\_master)  
  fk03\_user (user\_id on users\_master)**
* **Description: Tracks tools not returned on time or marked missing.**

**8. Table: tools\_transactions\_log**

* **Primary Key: log\_id**
* **Foreign Key: fk01\_tool (tool\_id on tools\_master)  
  fk03\_user (user\_id on users\_master)**
* **Description: Audit log table for appoint, return, update actions.**

**9. Table: roles\_master**

* **Primary Key: role\_id**
* **Foreign Key: Nil**
* **Description: Defines system roles (Admin, Toolskeeper, Maintenance User, Approver).**

**10. Table: user\_roles\_xref**

* **Primary Key: user\_id, role\_id**
* **Foreign Key: fk09\_role (role\_id on roles\_master)  
  fk02\_user (user\_id on users\_master)**
* **Description: Maps users to roles for RBAC implementation.**

**11. Table: tool\_category\_master**

* **Primary Key: category\_id**
* **Foreign Key: Nil**
* **Description: Tool categories for classification (e.g. Electrical, Mechanical).**

**12. Table: system\_parameters**

* **Primary Key: param\_key**
* **Foreign Key: Nil**
* **Description: Stores system-wide configurable parameters (alert thresholds, return durations).**

**🔧 Optional tables for advanced features**

**13. Table: password\_reset\_tokens**

* **Primary Key: token\_id**
* **Foreign Key: fk02\_user (user\_id on users\_master)**
* **Description: For password reset workflows with expiry timestamp.**

**14. Table: notifications**

* **Primary Key: notification\_id**
* **Foreign Key: fk02\_user (user\_id on users\_master)**
* **Description: In-app notifications for approvals, return due reminders.**

### Batch Processes:

**Several automated and semi-automated batch processes are executed daily by the Tools Tracker Admin and Toolskeeper to maintain data accuracy, track issued tools, and generate operational reports.**

**11.1.1 Batch for generation of daily tool issuance summary**

**At the end of each working day, a summary batch generates a daily issuance report based on unique combinations of:**

* **Tool issued**
* **Issued to (user or department)**
* **Quantity issued**
* **Date of issuance**
* **Issued by (Toolskeeper)**

**🔧 Executed by: Toolskeeper or Admin from the Daily Report page.**

**🔧Functionality:  
Fetches and aggregates issued tool records from the database (Appointed collection) and stores the summary data in a DailyIssuanceSummary collection for report viewing and audit.**

**11.1.2 Batch for generation of tool return data**

**After issuance summary generation, another batch process runs to generate tool return data. For each tool return recorded in the system:**

* **Updates tool availability in the ToolsMaster collection**
* **Logs return date, condition, and returned by user in Returns collection**
* **Updates tool status to ‘Available’ in real-time for subsequent issuances**

**🔧 Executed: Automatically upon submission of return forms by users or Toolskeeper approval.**

**11.1.3 Batch for overdue tools alert generation**

**At scheduled intervals (daily or hourly cron job), the system runs a batch process to:**

* **Identify tools whose return due date has passed and are not marked as returned**
* **Generate an overdue alert record in Alerts collection**
* **Send notifications via email or dashboard alerts to Toolskeeper and Admin for follow-up**

**🔧 Executed: Automatically by backend cron scheduler using node-cron or similar package.**

**11.1.4 Batch for month-end tools utilization report**

**At the end of each month, a batch generates MTD (Month-To-Date) and YTD (Year-To-Date) tools utilization reports:**

* **Calculates total quantity of each tool issued and returned**
* **Calculates average duration of tool usage per tool and per department**
* **Displays these figures in the Reports page for Admin analysis and operational planning**

**🔧 Executed by: Admin via the Reports module, or auto-triggered on last day of each month.**

**11.1.5 Batch for missing tools identification**

**A dedicated batch process runs to identify missing tools:**

* **Compares tool issued records with return records**
* **Flags tools not returned within X days (configurable parameter) as missing**
* **Updates their status to ‘Missing’ in the ToolsMaster collection**
* **Generates a missing tools report for management review**

**🔧 Executed: Automatically at daily scheduled time, managed by backend cron jobs.**

### User Interfaces:

**12.1 Master Creation**

**Several master creation and management screens are provided in the Tools Tracker system, mainly used by Admin users to set up tools inventory, manage user roles, and maintain data integrity. Below are the core master screens and their functionalities:**

**### 12.1.1 Tools Master [Screen Id – TT0001]**

**This screen is used to create, edit, or deactivate tools available in the system for issuance to users.**

**🔧 Mandatory fields:**

1. **Tool ID – Unique identifier for each tool. Auto-generated or manually assigned.**
2. **Tool Name – Descriptive name of the tool (e.g. ‘Spanner Set 14-32mm’).**
3. **Category – Dropdown selection (e.g. Hand Tool, Electrical Tool, Safety Equipment).**
4. **Stock Quantity – Total available quantity of the tool in stores.**
5. **Unit of Measure – Piece, Set, Kg, etc.**
6. **Status – Active / Inactive. By default, tools are created as Active. When tools are permanently disposed or discontinued, the Admin can mark them as Inactive, preventing further issuance requests.**

**🔧Notes:  
If a tool is marked as Inactive, it will be hidden from the Available Items page and issuance requests cannot be placed against it.**

**### 12.1.2 Users Master [Screen Id – TT0002]**

**The Users Master screen allows Admin to add, edit, delete users, or reset their passwords. Users can have different roles:**

* **Admin – Full access to all modules.**
* **Approver – Can approve/reject tool requests.**
* **Toolskeeper – Issues and returns tools, manages inventory updates.**
* **Maintenance User – Can request and return tools.**

**🔧 Mandatory fields:**

1. **User Email – Unique identifier and login email.**
2. **Password – Password set at creation (stored hashed in DB).**
3. **Role – Role assignment from dropdown (Admin, Approver, Toolskeeper, Maintenance User).**
4. **Status – Active / Inactive.**

**🔧 Functionality:**

* **Change password using inline button.**
* **Delete user (except Admin) via delete icon/button with confirmation prompt.**
* **Edit role dynamically.**

**### 12.1.3 Issuance & Return Entry [Screen Id – TT0003]**

**This screen is used by the Toolskeeper to appoint (issue) and return tools.**

**🔧 Issuance mandatory fields:**

1. **Tool ID – Selected from list of available tools.**
2. **Tool Name – Auto-filled based on Tool ID.**
3. **Quantity Issued – Must not exceed available stock.**
4. **Issued To (User Email) – Selected from registered users.**
5. **Date of Issue – Auto-filled with current date.**
6. **Promise Return Date – Expected return date.**

**🔧 Return mandatory fields:**

1. **Tool ID – Selected from list of appointed tools.**
2. **Return Date – Auto-filled with current date.**
3. **Condition – Good, Damaged, Missing (dropdown).**

**🔧Notes:  
Upon return, the tool’s stock is updated in the Tools Master table. If condition is marked as Damaged or Missing, Admin is alerted via the Missing Tools Report.**

**### 12.1.4 Approval Master [Screen Id – TT0004]**

**This screen is used by Approvers or Admin to:**

* **View all pending tool requests**
* **Approve or reject requests**

**🔧 Mandatory fields:**

1. **Tool Request ID – Auto-generated.**
2. **Requested By – User email.**
3. **Tool Name & Quantity – Auto-filled from request record.**
4. **Approval Status – Approved / Rejected (action button).**

**### 12.1.5 Reports & Alerts**

**All reports (e.g., Stock Report, Missing Tools Report, Master Info Report) are displayed in view-only pages built using React tables with sorting and export features:**

* **Daily Issuance Summary**
* **Tools Utilization (MTD/YTD)**
* **Overdue Tools Alerts**

**🔧ExcelExport:  
Users can export displayed tables to Excel using built-in export buttons implemented via xlsx or similar libraries in React frontend.**

**### 12.1.6 Alternative Search and Filter**

**For modules like Tools Master or Users Master, search and filter functionalities are provided:**

* **Admin can type part of tool name, user email, or category and view filtered results instantly.**
* **For example, typing “Hammer%” will display all tools starting with “Hammer”.**

**🔷 12.1.1 Tool Issuance Records [Screen Id – TT0005]**

**As mentioned in the Process Flow, each tool issuance record maintains details of which tool was appointed (issued), to whom, by which Toolskeeper, and with what expected return date. Records remain active until the tool is returned.**

**Functionality in Tools Tracker UI:**

**TopRegion(Viewonly):  
Displays tool master data (Tool ID, Tool Name, Available Stock).**

**Middle Region (Issuance Entry):**

**Here Toolskeeper enters issuance details:**

* **Issued To (User Email) – Selected from existing active users.**
* **Quantity Issued – Cannot exceed available stock.**
* **Promise Return Date – Mandatory, defines expected date of return.**
* **Condition at Issue – Dropdown (Good, Repaired, Minor Issue).**
* **Active Flag – By default, active. Becomes inactive when returned.**

**If a tool is reappointed before its return is recorded, the system throws an error stating *“Tool already issued and not yet returned.”***

**Bottom Region (Issuance History):**

**Displays all past issuance records for that tool with statuses (Active / Returned / Missing).**

**🔧 Mandatory Fields (Issuance Record)**

1. **Tool ID**
2. **User Email**
3. **Quantity Issued**
4. **Promise Return Date**
5. **Condition at Issue**

**🔴Validation:  
Tool issuance ID is unique. No duplicate issuance for same tool and user combination on the same date.**

**### 12.1.2 Tool Return Records [Screen Id – TT0006]**

**The Return Entry module records return details of tools issued earlier.**

**✅ Fields to enter:**

1. **Tool ID – Selected from list of issued tools.**
2. **Return Date – Auto-filled as current date.**
3. **Condition at Return – Good, Damaged, Missing.**

**🔧 Process:**

* **Upon return, tool stock is updated in Tools Master.**
* **If marked Damaged or Missing, an alert is generated in Missing Tools List for Admin to review and initiate recovery/replacement action.**

**🔷 12.2 Transaction Updates**

**### 12.2.1 Tool Request Approval [Screen Id – TT0007]**

**Functionality:  
Admins or Approvers view pending tool requests in the Approval screen:**

* **Approve: Tool is issued by Toolskeeper.**
* **Reject: Request status marked rejected with optional rejection note.**

**🔧 Mandatory Columns:**

1. **Request ID**
2. **Requested By (User Email)**
3. **Tool Name**
4. **Quantity Requested**
5. **Approval Status**

**### 12.2.2 Tool Condition Updates [Screen Id – TT0008]**

**Toolskeeper updates tool condition when returned:**

* **Good**
* **Needs Repair**
* **Missing**

**🔧Process:  
If marked as Needs Repair, the tool is flagged for maintenance and temporarily deactivated in stock availability. If Missing, Admin is alerted for disciplinary or replacement action.**

**### 12.2.3 Tool Maintenance Records [Screen Id – TT0009]**

**When tools require repairs:**

**Fields:**

1. **Tool ID**
2. **Issue Description**
3. **Maintenance Action Taken**
4. **Date of Repair**
5. **Status (Completed / Pending)**

**🔧Process:  
Tools under repair remain unavailable for issuance until marked Completed.**

**### 12.2.4 Tools Disposal Records [Screen Id – TT0010]**

**When tools are permanently damaged or obsolete:**

**✅ Fields:**

1. **Tool ID**
2. **Disposal Date**
3. **Reason for Disposal**
4. **Approved By (Admin)**

**🔧Process:  
Tool record is marked as Inactive and removed from available stock list.**

**### 12.2.5 Overdue Tools Monitoring [Screen Id – TT0011]**

**This alert module lists tools not returned by the promise return date.**

**✅ Fields:**

1. **Tool ID**
2. **User Email**
3. **Promise Return Date**
4. **Days Overdue**

**🔧Process:  
Admin or Toolskeeper follows up with the user to ensure immediate return or initiate recovery procedures.**

**### 12.2.6 Reports Generation [Screen Id – TT0012]**

**Admins can generate daily, monthly, or custom period reports for:**

* **Issuance summary**
* **Returns summary**
* **Missing tools list**
* **Tools under maintenance**
* **Tool utilization trends**

### Appendices

**13.1 Installation Requirements**

**Server Requirements (Backend)**

* **Node.js** (v18+ recommended)
* **MongoDB** (Atlas or local instance)
* **Express.js** (installed via project dependencies)

**Client Requirements (Frontend)**

* **React.js** (created via Vite or Create React App)
* **Tailwind CSS** (configured in project)
* **Axios** (for API calls)

**User Machine Requirements**

* Modern web browser (Chrome, Edge, Firefox)
* Internet connectivity to access deployed server OR LAN connectivity for local deployment
* For development:
  + Node.js installed
  + VSCode or similar IDE

**###Installation Steps**

1. **Clone the Repository**

bash

Copy code

git clone <repo-url>

cd tools-tracker

1. **Backend Setup**

bash

Copy code

cd backend

npm install

npm start

* Starts Express server on http://localhost:5000

1. **Frontend Setup**

bash

Copy code

cd frontend

npm install

npm run dev

* Starts React app on http://localhost:5173

1. **MongoDB Connection**

* Update your **.env** file in backend with:

ini

Copy code

MONGO\_URI=<your-mongodb-connection-uri>

PORT=5000

1. **Seed Admin & Toolskeeper Users**

* Run seeding script (if created) to insert default users.

**### 13.2 Supplementary Information**

* The application is fully **responsive** and works on mobile, tablet, and desktop.
* **Excel export** for reports is implemented using JS libraries like xlsx.
* No external proprietary runtime (e.g. Oracle Forms) is required.

**### 13.3 References**

* MERN stack official documentation:
  + [MongoDB Docs](https://www.mongodb.com/docs/)
  + [Express.js Docs](https://expressjs.com/)
  + [React.js Docs](https://react.dev/)
  + Node.js Docs
* Tailwind CSS – [tailwindcss.com/docs](https://tailwindcss.com/docs)
* Axios – axios-http.com/docs



**---- END OF DOCUMENT ---**