# Software Requirements Specification

# for

# ATTRAC

# Version 1.0

Prepared by Aadil Faisal, Suvam Poddar, Shrvan Sudhakara and Anusha Venkatramanan

Manipal Institute of Technology

2nd July 2024

Table of Contents

Table of Contents [ii](#__RefHeading___Toc441230970)

Revision History [ii](#__RefHeading___Toc441230971)

1. Introduction [1](#__RefHeading___Toc441230972)

1.1 Purpose [1](#__RefHeading___Toc441230973)

1.2 Intended Audience and Reading Suggestions [1](#__RefHeading___Toc441230975)

1.3 Product Scope [1](#__RefHeading___Toc441230976)

1.4 References [1](#__RefHeading___Toc441230977)

2. Overall Description [1](#__RefHeading___Toc441230978)

2.1 Product Perspective [1](#__RefHeading___Toc441230979)

2.2 Product Functions [2](#__RefHeading___Toc441230980)

2.3 User Classes and Characteristics [2](#__RefHeading___Toc441230981)

2.4 Operating Environment [2](#__RefHeading___Toc441230982)

2.5 Design and Implementation Constraints [3](#__RefHeading___Toc441230983)

2.6 Assumptions and Dependencies [3](#__RefHeading___Toc441230985)

3. External Interface Requirements [4](#__RefHeading___Toc441230986)

3.1 User Interfaces 4

3.2 Hardware Interfaces [5](#__RefHeading___Toc441230988)

3.3 Software Interfaces [5](#__RefHeading___Toc441230989)

3.4 Communications Interfaces [5](#__RefHeading___Toc441230990)

4. System Features [6](#__RefHeading___Toc441230991)

4.1 Details Module [5](#__RefHeading___Toc441230992)

4.2 Add Transaction Module [6](#__RefHeading___Toc441230993)

4.3 View Transaction Module [6](#__RefHeading___Toc441230988)

4.4 Functional Requirements [7](#__RefHeading___Toc441230988)

5. Other Nonfunctional Requirements [9](#__RefHeading___Toc441230994)

5.1 Performance Requirements [9](#__RefHeading___Toc441230995)

5.2 Safety Requirements [9](#__RefHeading___Toc441230996)

5.3 Security Requirements [9](#__RefHeading___Toc441230997)

5.4 Software Quality Attributes [9](#__RefHeading___Toc441230998)

5.5 Business Rules [10](#__RefHeading___Toc441230999)

Appendix : Analysis Models 11

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

ATTract (Advanced Transaction Tracker) aims to provide a comprehensive transaction tracking service to users. This software will enable users to track the incoming and outgoing transactions in real-time, providing accurate and up-to-date information on all the transactions made. The package tracking service will include various features such as transaction status updates, time of transactions, and proof of delivery. This software will be designed to provide a user-friendly and intuitive interface, ensuring that users can easily navigate and utilize the service. Overall, ATTract's goal is to provide a reliable and efficient transaction tracking service that meets the needs and expectations of its users.

## Intended Audience and Reading Suggestions

The intended audience for the documentation of ATTract could be developers, software engineers, and programmers who have a basic understanding of Python programming language and SQL databases.

For reading suggestions, refer to 1.4.

## Product Scope

*ATTract, short for Package-Tracker, is a package tracking service that helps the user to keep track of their existing placed order.*

## References

* <https://www.djangoproject.com/>
* <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>
* <https://developer.mozilla.org/en-US/docs/Web/HTML>
* <https://developer.mozilla.org/en-US/docs/Web/CSS>
* <https://docs.python.org/>

# Overall Description

## Product Perspective

ATTract, short for Advanced Transaction Tracker, is a transaction tracking service designed for tracking incoming and outgoing transactions. This service helps users keep track of their financial activities efficiently. We utilized SQL for the database, ensuring robust data management and storage. The front-end of the website is developed using HTML and CSS, providing a user-friendly and responsive interface. For integration and management, we employed Django, a powerful Python-based web framework, to seamlessly connect the frontend and backend, enabling smooth and efficient transaction tracking.

## Product Functions

* CRUD Operations for every table
* Sorted and specific viewing of the data

## User Classes and Characteristics

Based on the description of ATTract, the following user classes can be identified:

* Students: These are individuals enrolled in the institution who will use the application to track their own transactions related to fees, scholarships, and other financial activities. Students may have varying levels of technical expertise and will need access to view transactions, add transactions, and edit their personal details.
* Parents: These are guardians of the students who will use the application to monitor and manage their child's financial transactions at the institution. Parents may use the application occasionally or frequently based on the transaction activities of their children. They will need access to view transactions, add transactions, and edit their personal details.
* Faculty: These are the teaching and administrative staff of the institution who will use the application to track and manage transactions related to their professional activities. Faculty members will need access to view transactions, add transactions, and edit their personal details.
* Developers: These are individuals who will be responsible for maintaining and updating the application. They will require a deep understanding of the system's architecture, database management, and software development principles.

The most important user class for ATTract is the Students, who are the primary end-users of the application and the main beneficiaries of its functionalities, making their user experience crucial. Parents are significant for monitoring and managing the financial aspects of their children’s education, helping to ensure that transactions are kept up-to-date and accurate. Faculty members are important for managing their professional financial transactions and ensuring the smooth operation of their respective activities within the institution. Developers are essential for the ongoing maintenance, updates, and overall management of the application, ensuring the system remains functional, secure, and up to date with the latest features and fixes.

## Operating Environment

To describe the environment in which ATTract will operate, we must consider the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist:

**Hardware Platform:**

The software is designed to operate on a range of hardware platforms, including desktop and laptop computers, tablets, and mobile devices. It should be compatible with any device that supports the required operating system and meets the minimum system requirements for web applications.

**Operating System:**

The software is designed to be compatible with multiple operating systems, including Windows, macOS, and Linux. It should be tested and verified to work on the latest versions of these operating systems, including Windows 10, macOS Mojave or later, and Ubuntu 18.04 or later.

**Software Components:**

The software requires several software components to operate: SQL is used for the database management system, HTML and CSS for the frontend design and user interface, and Django, a powerful Python-based web framework, for backend development and integration. The application should be compatible with modern web browsers, including the latest versions of Chrome, Firefox, Safari, and Edge.

## Design and Implementation Constraints

There are several items or issues that may limit the options available to the developers of ATTract. These include:

* Corporate or regulatory policies: The developers must comply with any corporate or regulatory policies that affect the development of the software. This may include restrictions on the use of certain technologies or requirements for data privacy and security.
* Hardware limitations: The developers must take into account any hardware limitations that may affect the performance or functionality of the software. If the software must run on low-end hardware, this may limit the amount of memory or processing power that can be used.  
    
  **Hardware Requirements**:

Processor: Intel Core 2 Duo or above

Processor Speed: 1.5GHZ or above

RAM: 4 GB RAM or above

Hard Disk: 20 GB hard disk or above

* Interfaces to other applications: If the software must interface with other applications, the developers must ensure that these interfaces are designed and implemented correctly. They must also consider any limitations or requirements of the other applications.
* Specific technologies, tools, and databases: If the customer has specified specific technologies, tools, and databases to be used, the developers must adhere to these requirements. This may limit the options available for the development of the software.
* Design conventions or programming standards: If the customer's organization will be responsible for maintaining the delivered software, the developers must adhere to any design conventions or programming standards specified by the customer. This may limit the options available for the development of the software.

## Assumptions and Dependencies

There are several assumed factors that could affect the requirements stated in the SRS for ATTract. These include:

* Third-party or commercial components: The SRS assumes that the third-party or commercial components used in the development of the software will be available and compatible with the operating environment. If these assumptions are incorrect or the components become unavailable, it may impact the development of the software.
* Development or operating environment: The SRS assumes that the development or operating environment will be available and suitable for the development of the software. Any changes or issues with the environment may affect the development and delivery of the software.
* Constraints: The SRS assumes that any constraints, such as time, budget, or resource limitations, will be managed effectively. Any changes to these constraints may affect the requirements and delivery of the software.
* Dependencies on external factors: The SRS assumes that any external factors, such as software components that are reused from another project, are available and compatible with the operating environment. Any changes or issues with these external factors may affect the development and delivery of the software.
* Compatibility with future software updates: The SRS assumes that the software will be compatible with future updates to the operating system, database, or third-party components used in the software. If these assumptions are incorrect, it may affect the usability and functionality of the software in the future.
* User behavior: The SRS assumes that users will use the software as intended and will provide accurate input. Any unexpected user behavior or inaccurate input may affect the functionality of the software.
* Scalability and performance: The SRS assumes that the software will be scalable and performant for the intended user base. Any changes in user behavior or an increase in the number of users may affect the scalability and performance of the software.

# External Interface Requirements

## User Interfaces

*Dashboard screen with these modules as the MODULES:*

1. *Choosing page:*
   1. *Student*
   2. *Parent*
   3. *Faculty*

*2. Signup Page*

*3. Operations page:*

*- Edit Details Module*

*- Add Transaction Module*

*- View Transaction Module*

*Each of which has its own appropirate screens*

## Hardware Interfaces

ATTract doesn’t come with a direct hardware integration with it, but can be integrated with mobile devices and other external peripherals like barcode scanner.

## Software Interfaces

Database: SQL

Language: Python, HTML, JavaScript

GUI: CSS

Database Adapter: Django

## Communications Interfaces

*ATTract uses various communication interfaces for its functioning, which are:*

* Network server communication Protocols: Only local host as of now, but can use FTP and HTTP later on.
* Electronic forms.

# System Features

## Login Module

4.1.1 Description and Priority

This feature allows users to login as per their role. High priority.

4.1.2 Stimulus/Response Sequences

* User opens the Choosing page module.
* The system displays the three options to login as – student, parent and faculty.
* The user selects one of the options.
* The system displays the corresponding module signup/login page
* The user performs necessary actions and logs in using their valid credentials.

4.1.3 Functional Requirements

The system should provide an interface to access the student, faculty and parent roles before login.

The system should allow users to login with their credentials after selecting their roles.

The system should allow users to provide their valid credentials and allow them to create new credentials if they are new.

The system should validate user inputs to prevent data errors.

The system should provide error messages if an error occurs during data entry.

The system should allow users to search for specific inventory data using filters.

The system should be able to generate reports on inventory data as required.

## Edit Details Module

4.2.1 Description and Priority

*This feature allows users to manage user and contact information. Medium priority.*

4.2.2 Stimulus/Response Sequences

* + - User opens the Edit Details Module.
    - The system displays the corresponding module with its features.
    - The user performs necessary actions such as adding, editing, or deleting the data.

4.2.3 Functional Requirements

The system should provide an interface to access the Edit Details module.

The Edit Details module should allow users to add, edit, or delete user information with details

such as userID, name, address, and contact information.

The system should validate user inputs to prevent data errors.

The system should provide error messages if an error occurs during data entry.

The system should allow users to search for specific customer or contact data using filters.

The system should be able to generate reports on customer or contact data as required.

## Administration Module

4.3.1 Description and Priority

*This feature allows administrators to manage user, address, carrier, route, and delivery route information. High priority..*

4.3.2 Stimulus/Response Sequences

* + - User opens the Administration Module.
    - The system displays the User, Address, Carrier, Route, and Delivery Route modules.
    - The user selects one of the modules.
    - The system displays the corresponding module with its features.
    - The user performs necessary actions such as adding, editing, or deleting the data.

4.3.3 Functional Requirements

The system should provide an interface to access the User, Address, Carrier, Route, and Delivery

Route modules.

The User module should allow administrators to add, edit, or delete user information with.

**4.4 Functional Requirements:**

**4.4.1 User Management Module**:

**Requirement 1: User Role Selection**

The system shall allow users to register with their name, email, and password.

The system shall verify that the email address is valid and not already in use.

The system shall store the user's information in the Users table.

**Requirement 2: User Login**

The system shall allow registered users to login using their email and password.

The system shall verify the user's email and password against the stored information in the Users

table.

The system shall display an error message if the login information is incorrect.

**Requirement 3: User Signup**

The system shall allow users to sign up using their email and OTP received in their mail.

The system shall verify that the email is valid and not already in use.

The system shall store user information in the Users table.

The system shall display an error message if the email is already in use.

**Requirement 4: User Profile Management**

The system shall allow users to view and edit their profile information, including name and password.

The system shall verify that the email address is valid and not already in use.

The system shall update the user's information in the Users table.

**Requirement 5: User Authorization**

The system shall allow users with administrative privileges to perform administrative tasks, such as

creating and editing users.

**4.4.2 Operations Module:**

**Requirement 1: Add/Edit Details**

The system shall allow users to add or edit their personal details, including name, parent name, address, and other contact information.

The system shall store user details information in the Users table.

The system shall provide a button to submit the information.

**Requirement 2: Add Transaction**

The system shall allow users to add a transaction with fields for receiver ID, transaction amount, and receiver name.

The system shall interpret a receiver ID of 10000 as a transaction to the institution.

The system shall store transaction information in the Transactions table.

The system shall provide a button to submit the transaction.

**Requirement 3: View Transactions**

The system shall display all past transactions done by the user in a tabular format.

The system shall show outgoing and incoming transactions, including receiver name and ID as well as the transaction amount.

The system shall retrieve transaction information from the Transactions table.

**4.4.4 Administrative Functions**

**Requirement 1: Administrative Tasks**

The system shall allow users with administrative roles to manage user accounts and transaction records.

The system shall include functionalities for adding, editing, and deleting user and transaction records.

# Other Nonfunctional Requirements

## Performance Requirements

* The application must handle a large volume of data and user requests efficiently.
* Response time should be less than 3 seconds for all actions, including data retrieval, update, and deletion.
* The application should process and update all data in real-time to ensure a smooth and efficient user experience.

**Rationale:** Ensuring quick response times and real-time processing is crucial for maintaining user satisfaction and operational efficiency, especially in an environment with high data volumes.

.

## Safety Requirements

* The application must ensure the safety and integrity of all user data.
* Safeguards must be in place to prevent unauthorized access or manipulation of data.
* Actions that could result in data loss or damage should be prevented.
* The application must comply with relevant safety policies and regulations and obtain necessary safety certifications.

**Rationale:** Protecting user data and maintaining data integrity is essential to build trust and comply with legal standards.

## Security Requirements

* The application must ensure the security and privacy of all user data.
* User identity authentication must be implemented to prevent unauthorized access.
* All data must be protected against security threats.
* The application must comply with relevant security policies and regulations and obtain necessary security and privacy certifications.

**Rationale:** Ensuring data security and privacy is critical to protect user information and comply with regulatory requirements.

## Software Quality Attributes

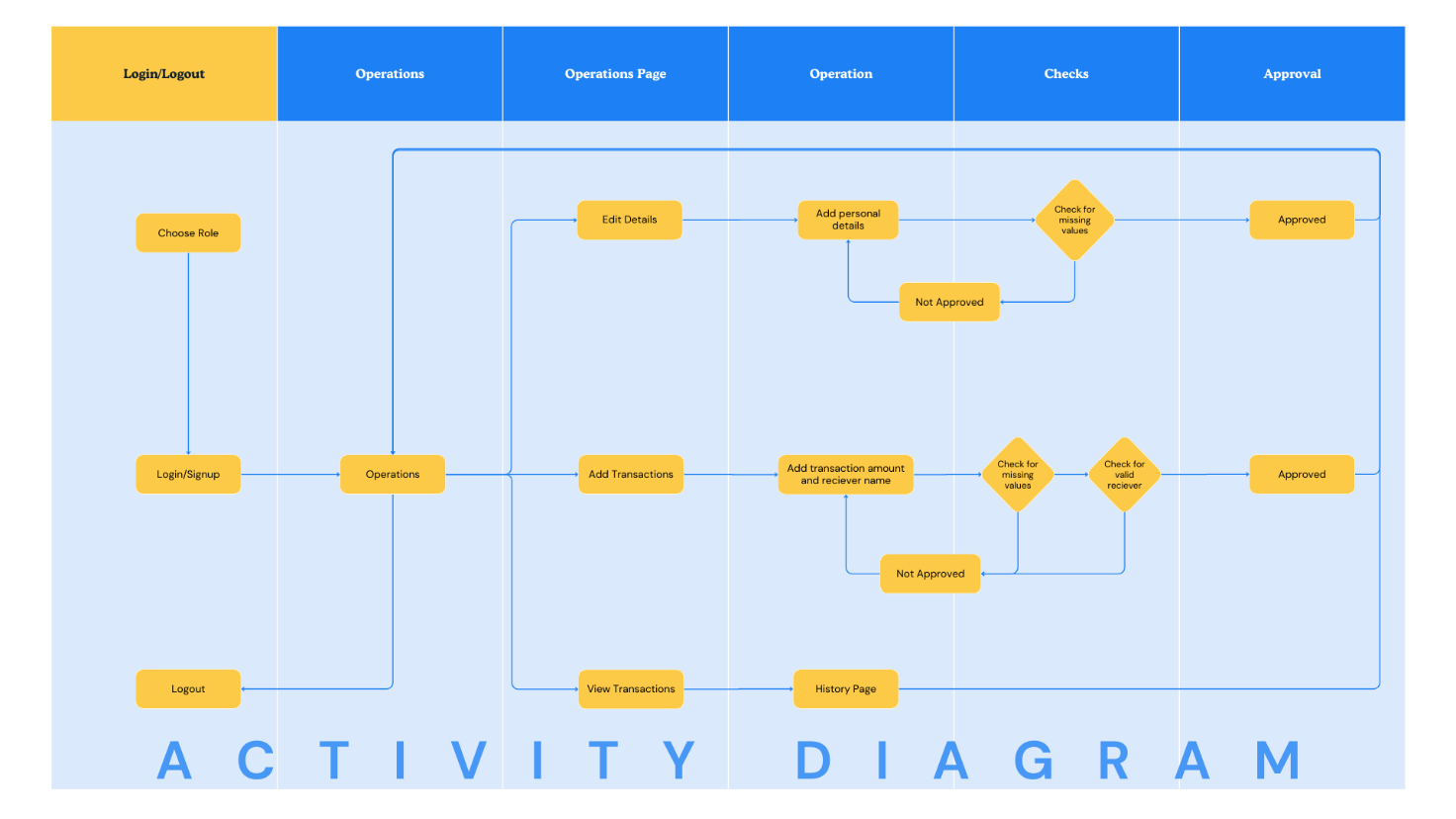
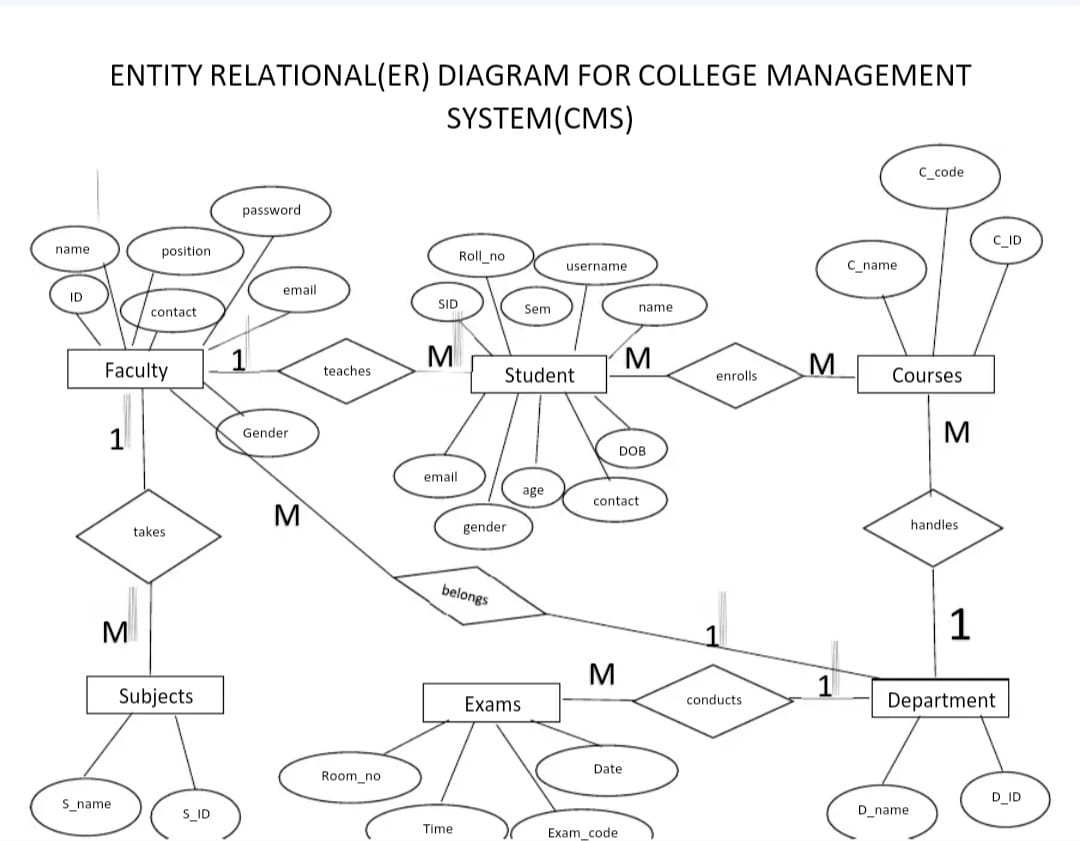
* ***Maintainability:*** *The application must be easy to maintain and update, with clear and concise documentation for all modules.*
* ***Reliability:*** *The application must ensure that all data is accurate and up-to-date.*
* ***Usability:*** *The application should be easy to use and navigate, with an intuitive user interface that requires minimal training.*

***Rationale:*** *High-quality software that is easy to maintain, reliable, and user-friendly will improve user satisfaction and reduce operational costs.*

## Business Rules

* The application must enforce operating principles and rules outlined in the project description.
* Users must have specific roles and permissions to perform certain functions within the system.
* The application must enforce these rules and prevent unauthorized actions that violate them.

**Rationale:** Enforcing business rules and role-based permissions ensures proper governance, operational consistency, and compliance with institutional policies.

Appendix: Analysis Models