SOFTWARE REQUIREMENTS SPECIFICATION

for

Haat Baran

Prepared by

Mostofa Hasin Mahdi (2223071109)
Talal Sharar Apurbo (223071108)

Submitted to

Md. Akram Hossain
Lecturer
Department of CSE and CSIT
Shanto-Mariam University of Creative Technology

October 31, 2025

Contents

1 Preface

This Software Requirement Specification (SRS) describes the requirements for the Haat Baran project , a digital funding and verification platform for underprivileged individuals seeking startup capital to begin small businesses.
The purpose of this document is to define all functional and non-functional requirements of the system, serving as an agreement between developers, stakeholders, and future contributors. It will guide the development team and provide a solid base for further maintenance or upgrades.
Version 1.0 of Haat Baran aims to introduce a reliable and transparent system with strong verification measures and on-site profiling by field volunteers. Donors will experience a secure, user-friendly interface for browsing verified applicants and providing financial support safely.

2 Introduction

A Software Requirement Specification (SRS) is a comprehensive document that outlines the purpose, scope, functionality, and constraints of a software system to be developed. It serves as a bridge between the client's expectations and the developer's understanding, ensuring that the final product aligns perfectly with user needs and technical feasibility. The SRS also acts as a foundation for design, testing, maintenance, and future upgrades of the software.

The project titled "Haat Baran" aims to develop a community-driven micro-funding platform that empowers individuals living below the poverty line to start small businesses and become financially independent. The system will connect poor individuals — who often lack access to traditional banking or government assistance — with donors and philanthropists who are willing to provide them with financial support.

In many developing regions, poverty is not only a result of lack of opportunity but also a lack of access to verified funding channels. Most existing donation systems are unstructured and prone to misuse, while many underprivileged people are unable to participate because they do not possess formal identification or digital literacy. The Haat Baran system aims to close this gap by introducing a transparent and verifiable digital platform where donors can confidently contribute, knowing that their funds reach genuine and deserving individuals.

The platform will consist of three primary interfaces:

1. Volunteer Application: Used by trained field workers to visit applicants, collect their details, photos, and location, and submit this information to the central system.
2. Donor Application / Web Portal: Designed for donors to view verified profiles of applicants, check their background and business goals, and securely contribute funds using digital payment gateways.
3. Administrator Dashboard: A web-based control panel for approving applicants, monitoring funding flow, generating reports, and ensuring transparency.

Through this system, poor individuals can receive seed funding to start small ventures such as tea stalls, tailoring shops, or grocery kiosks, enabling them to become self-sufficient. Meanwhile, donors and organizations can track how their funds are utilized and view progress reports over time, building trust in the process.

The Haat Baran project not only addresses financial inequality but also introduces accountability, accessibility, and digital empowerment in the social aid ecosystem. It promotes sustainable development by converting donations into long-term livelihood opportunities.

3 Glossary

The glossary provides a list of key technical terms and concepts used throughout this Software Requirement Specification (SRS) for Haat Baran. Its purpose is to ensure a clear and consistent understanding of the terminology. Each term listed here is defined precisely to avoid ambiguity.

Table 3.1: Glossary

Technical Term	**Description**
Applicant	An underprivileged person seeking financial help to start a business
Donor	A person who contributes funds through the system to verified applicants
Volunteer	A field agent who collects applicant data
Admin Panel	Centralized web dashboard for monitoring, approval, and management
Verification	The process of validating an applicant's authenticity via biometric and photographic data
Location Info	Location data marking to ensure authenticity of applicant's location
Payment	Digital System that allows monetary transactions
Cloud Database	Central Server for securely storing applicant and donor data
Encryption	Method used to secure sensitive user data
Profile	Digital record containing applicant's identity, verification, and funding details

4 User Requirements Definition

Requirements describe what the software must achieve to meet user and client expectations. Derived from client meetings, the requirements for Haat Baran are divided into two categories: Functional Requirements, which define the system's core operations, and Non-Functional Requirements, which describe its quality attributes and constraints. These specific user requirements are enumerated in the following table.

Table 4.1: Definition of User Requirements

Requirement Type	**Definitions of Requirement**
Functional Requirements	1. Field volunteers will register applicants and collect their information.
	2. Volunteers will insert applicant name, photo, fingerprint photo and location.
	3. Admins will verify applicant data through the web panel.
	4. Donors will view verified applicants and donate securely.
	5. The system will send notifications about funding status to users.
	6. Admins can approve or reject applications based on verification.
Non-Functional Requirements	1. The system must be secure, encrypted with end-to-end encryption.
	2. The interface must be easily usable by individuals with limited technical literacy.
	3. The application should run smoothly on Android phones.
	4. The system must support multiple concurrent users.
	5. The server must ensure data integrity and high uptime.
	6. The application should require minimal internet bandwidth.

5 System Architecture

The Haat Baran system is built on a client-server architecture, structured around three core modules: a mobile Volunteer App for field data collection and applicant registration; a Donor App/Web Portal for viewing verified applicants and processing donations; and a web-based Admin Panel for comprehensive data management, verification, and report generation. The system architecture of Haat Baran is depicted in the figure below:

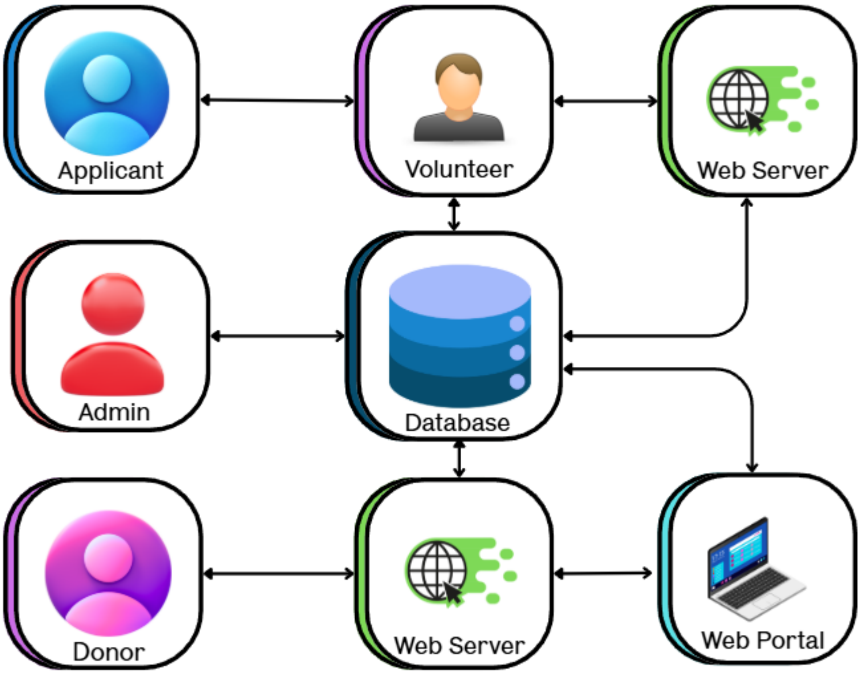

Figure 5.1: System Architecture

6 User Requirements Specification

This Software Requirements Specification (SRS) document defines the essential requirements for the Haat Baran project. These requirements were developed to provide both the development team and the client with a clear, unambiguous understanding of the system to be built. A complete list of the detailed user requirements can be found in the following Table.

Table 6.1: Specification of User Requirements

Requirement Type	**Specification of Requirements**
Functional Requirement	1.1 Volunteers can log in to the app using secure credentials.
	1.2 Volunteers fill up applicant's name, age, gender, and family data.
	1.3 Volunteers capture biometric data and photographs.
	2.1 Admin reviews and verifies the submitted information.
	2.2 Verified profiles become visible to donors on the donor portal.
	3.1 Donors can browse, filter, and donate to applicants.
	3.2 Donors receive confirmation and transaction receipts.
	3.3 Admins can track total donations and generate reports.
Non-Functional Requirement	1.1 All sensitive data must be stored in encrypted format.
	1.2 System must authenticate every actor (Volunteer, Donor, Admin).
	2.1 The response time for each transaction should be under 2 seconds.
	2.2 System should support at least 500 concurrent users.
	2.3 Biometric Scanner and facial recognition should be done within 5 seconds.
	3.1 The software should operate on devices with minimum 1GB RAM and 1GHz CPU.
	3.2 Interface should be visually clean, bilingual (Bangla/English), and intuitive for all age groups.

7 System Model

7.1 Use Cases

Use cases describe the interaction between the user and the system to achieve specific goals. Use cases are needed to understand how the entire system works. The following table lists the key use cases of the Haat Baran system.

Table 7.1: List of Use Cases

Use Case	Title
UC1	Applicant Registration
UC2	Volunteer Login
UC3	Applicant Verification and Approval
UC4	Donor Registration and Login
UC5	View and Fund Verified Applicants
UC6	Donation Request
UC7	Send Funding Notification
UC8	Generate Report and Progress Data

The use cases are described in details as follows:

7.1.1 UC1: Applicant Registration

Actors:
1. Volunteer
2. Applicant
3. System

Preconditions:
1. Volunteer is logged into the Haat Baran mobile app.
2. Device is connected to the Internet or has offline storage enabled.
3. Volunteer has applicant consent for registration.

Main Success Scenario:
1. Volunteer clicks “Register New Applicant.”
2. Fills in personal, family, and business details.
3. Captures applicant photo, fingerprint, and location data(Upazilla/union).
4. Clicks “Submit” to upload the data to the storage.

5. System confirms successful data upload.

Postcondition:
Applicant profile is created and stored in the central database awaiting admin verification.

Alternative Course:
3.a. No internet connection → Data stored locally and synced later.
4.a. Fingerprint capture fails → Prompt volunteer to retry.

7.1.2 UC2: Volunteer Login

Actors:

1. Volunteer
2. System

Preconditions:
1. App installed on device.
2. Volunteer has valid credentials.
Internet connection is active.

Main Success Scenario:

1. Volunteer opens the app.
2. Enters username and password.
3. Clicks “Login.”
4. System verifies credentials and grants access.

Postcondition:
Volunteer dashboard is displayed showing applicant registration options.

Alternative Course:
3.a. Incorrect credentials → Display “Invalid Login” message.
3.b. Forgotten password → Click “Forgot Password” to reset.

7.1.3 UC3: Applicant Verification and Approval

Actors:
1. Administrator
2. Volunteer
3. System

Preconditions:
1. Applicant data submitted by volunteer.
2. Admin logged into web dashboard.
3. Internet connectivity available.

Main Success Scenario:

1. Admin opens pending applicant list.
2. Reviews submitted biometric and photo data.
3. Confirms GPS location and authenticity.
4. Approves or rejects application.

Postcondition:
Approved applicants are marked "Verified" and become visible to donors.

Alternative Course:
2.a. Missing data → Request volunteer resubmission.

7.1.4 UC4: Donor Registration and Login

Actors:
1. Donor
2. System

Preconditions:
1. Donor has valid email or phone number.
2. Internet connection available.

Main Success Scenario:
1. Donor clicks "Create Account."
2. Fills registration form with contact and payment details.
3. Logs in successfully.

Postcondition:
Donor dashboard is displayed showing verified applicants.
Alternative Course:

2.a. Duplicate email → Prompt to use a different email.
3.a. Wrong Password → Try again.

7.1.5 UC5: View and Fund Verified Applicants

Actors:
1. Donor
2. System

Preconditions:
1. Donor is logged in.
2. Applicant profiles are verified by admin.
3. Backend is online

Main Success Scenario:
1. Donor opens “Browse Applicants.”
2. System displays verified applicant list.
3. Donor selects an applicant.
4. Views profile details and funding goal.
5. Clicks “Donate now.”

Postcondition: Must have internet connection.

Alternative Course:
3.a. Profile not loading → Retry after refreshing connection.

7.1.6 UC6: Process Donation Payment

Actors:
1. Donor
2. Donor portal system

Preconditions:
1. Donor logged into the app.
2. Backend is active and connected.

Main Success Scenario:
3. Donor clicks “Confirm Donation.”
4. System sends donation request to admin.
5. Admin confirms donation request.
6. Donor proceeds to donate through their comfortable payment system.
7. Transaction information is saved.

Postcondition:

Donor receives confirmation message and receipt.
Applicant's funding progress updated.

Alternative Course:

3.a. Transaction failed → Prompt user to retry with valid details.

7.1.7 Sending Fund Notification

Actors:
1. System
2. Applicant
3. Donor

Preconditions:
1. Donation successfully processed.
2. Applicant and donor linked via transaction.

Main Success Scenario:
1. System automatically sends a notification to applicant confirming donation.
2. Donor receives confirmation of successful funding.
3. Admin dashboard updates funding completion percentage.

Postcondition: Both donor and applicant are notified of transaction status.

Alternative Course:
1.a. Notification fails → Retry sending after a short interval.

7.1.8 UC8: Generate Report and Progress Data

Actors:
1. Administrator
2. System

Preconditions:
1. Database has active applicant and donation records.
2. Admin is logged in.
3. Main Success Scenario:
4. Admin clicks "Generate Report."
5. System compiles donor and applicant data.

6. Displays funding statistics, success rates, and pending verifications.

Postcondition: Report is successfully generated.

Alternative Course:
2.a. Server busy → Display "Please try again later."

7.2 Use Case Diagram

The use case diagram shows the interaction between applicant, volunteer, donor, admin and the system graphically. The use case diagram is shown in the following figure.

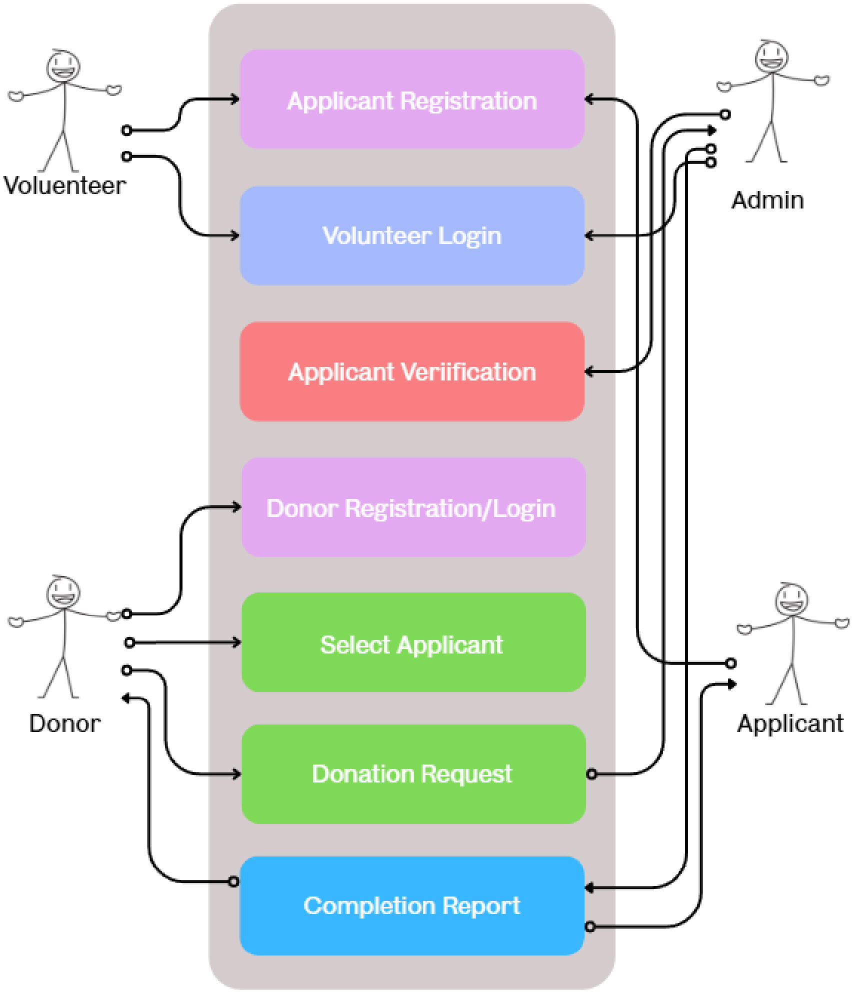

Figure 7.1: Use Case Diagram

8 System Evolution

Software evolution refers to the continuous process of improving and adapting the system to meet new requirements, fix issues, or enhance performance. The Haat Baran system is designed to be flexible, upgradable, and scalable according to user demand and technological changes.

The evolution of the Haat Baran platform will include:
Version 1.0: Core functions — volunteer registration, applicant verification, donor funding, and admin control panel.
Version 2.0: Introduction of post-funding tracking and automated progress reports for donors.
Version 3.0: Expansion with AI-driven verification assistance, fraud detection, and predictive funding suggestions.

The system is implemented using modular architecture, making it easy to add or modify components without redesigning the entire structure. It is optimized to handle growing user bases by supporting cloud scaling, ensuring smooth operation even as data and transaction volumes increase.

Performance benchmarks (with moderate hardware):
Up to 500 concurrent user requests per minute.
Supports real-time updates for funding progress.
Data synchronization for offline volunteer app usage.

The system is designed to evolve efficiently with improvements in cloud infrastructure, biometric technology, and data analytics capabilities as well as AI enhancements.

9 Appendices

Appendices contain technical specifications and supporting details for hardware and database configurations used in the Haat Baran system.

9.0.1 Appendix A: Hardware Specification

The system is developed and hosted on a cloud-based server setup equivalent to mid-level enterprise configurations.

Table 9.1: Server Specification

Processor	Intel® Xeon® Silver 4310
Number of Processors	1
Processor Core Available	12
Processor Cache	8MB (1 x 8MB) Level 3 cache
Processor Speed	3.3GHz
Chipset	Intel® C236 Chipset
Power Supply Type	300W Multi-Output Power Supply
Memory	4GB DDR4
Memory Slots	4 DIMM slots
Memory Type	1R x8 PC4-2133P-E-15
Memory Protection Features	Un-buffered ECC
Installed Hard Drives	LFF SATA: 1TB
Maximum Internal Storage	24TB
Optical Drive Type	SATA 9.5mm DVD RW
System Fan Features	Non-Pluggable Fan
Network Controller	Intel® Ethernet Connection I219-LM
Storage Controller	Integrated SATA RAID
Infrastructure Management	Intel® Active Management Technology (Intel® AMT 11.0)

9.0.2 Appendix B: Database Specification

The Haat Baran platform uses MongoDB as the database management system

Table 9.2: Database Specification

Property	**Specification**
Database Type	Relational
Architecture	Client Server Model
OS Support	Linux, Windows, macOS
Access Control	Role based access - Admin, Volunteer, Donor
Backup Frequency	Daily incremental and weekly full
Maximum Storage Capacity	Up to 20TB
Maximum Concurrent Queries	200+
Indexing	B-tree, Hash, Full text
Partitioning	Supported
Security Features	SQL injection prevention, audit logging, user access control

10 Index

10.0.1 A. List of Figures

Figures are graphical representation of information. The figures used in this document is listed in the following table

Table 10.1: List of Figures

10.0.2 B. Alphabetic

11 Contribution

Table 11.1: Contribution table

Name	Contribution
Mostofa Hasin Mahdi	Gathering Resources for system design and architecture
Talal Sharar Apurbo	Gathering resources for user requirements specification