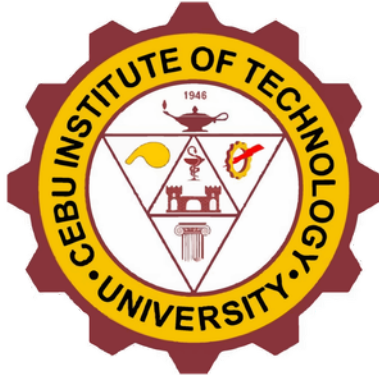


CEBU INSTITUTE OF TECHNOLOGY UNIVERSITY



COLLEGE OF COMPUTER STUDIES

Software Requirements Specifications

for

“Tarabaho: Tara Trabaho”



Developers:

Batucan, Derick Wayne A.

Bacusmo, Vic Andre D.

Quieta, Angelo C.

Change History

Table of Contents

1. Introduction	5
1.1. Purpose	5
1.2. Scope	5
Key Features & Functionality	6
Limitations and Restrictions	6
What the Software Will Not Do	7
Application & Benefits	7
1.3. Definitions, Acronyms and Abbreviations	7
1.4. References	7
2. Overall Description	8
2.1. Product perspective	8
Modular Decomposition	9
2.2. User characteristics	10
1. Worker / Trabahador	10
2. User / Client	10
3. Administrator	11
2.4. Constraints	12
2.5. Assumptions and dependencies	14
3. Specific Requirements	15
3.1. External interface requirements	15
3.1.1. Hardware interfaces	15
3.1.2. Software interfaces	15
3.1.3. Communications interfaces	16
3.2. Functional requirements	16
Web Application Function	16
Mobile Application Function	16
3.4 Non-functional requirements	17
Performance	17
Security	17
Reliability	17

1. Introduction

1.1. Purpose

- Describe the purpose of the SRS;
- Specify the intended audience for the SRS.

The purpose of this **Software Requirements Specification (SRS)** document is to outline the functional and non-functional requirements for **TaraTrabaho**—a job-matching platform that connects individuals seeking temporary or short-term work with those in need of services. Additionally, the platform aims to provide opportunities for **teenagers aged 15 and above** to gain work experience, develop essential life skills, and earn income in a safe and legal environment.

TaraTrabaho is designed to be an **efficient, reliable, and accessible** solution for hiring on-demand **Trabahadors** for various tasks, whether short-term or long-term. It not only benefits employers in finding quick assistance but also empowers job seekers by offering **flexible work opportunities**.

This document serves as a **guide** for all stakeholders involved in the project's development, ensuring that the application's technical and functional aspects are well-documented, aligning expectations, and streamlining the development process.

Intended Audience

This document is intended for the following stakeholders:

- **Project Owners & Developers** – To understand and implement the system requirements.
- **Designers & UI/UX Team** – To align the user interface with functional needs.
- **Testers & Quality Assurance (QA) Team** – To verify that all requirements are met.
- **Investors & Business Stakeholders** – To assess the feasibility and potential impact of the application.
- **End Users (Job Seekers & Employers)** – To ensure the platform meets their needs effectively.

1.2. Scope

- Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.);
- Explain what the software product(s) will, and, if necessary, will not do;
- Describe the application of the software being specified, including relevant benefits, objectives, and goals;
- Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist.

Product Overview

Tarabaho is a job-matching platform designed to connect individuals seeking temporary, short-term, or task-based work with those in need of services. The platform facilitates a streamlined hiring process, allowing users to search, hire, and manage Trabahadors efficiently.

Tarabaho is available as both a web and mobile application, ensuring accessibility across different devices. The system supports various work categories, such as household assistance, errands, tutoring, cleaning, maintenance, and more, providing flexible employment opportunities, especially for teenagers and individuals looking for extra income.

Key Features & Functionality

Job Posting & Search – Users can post jobs or browse available Trabahadors based on skills, location, and availability.

Worker Profiles & Reviews – Job seekers can create profiles showcasing their experience, skills, and ratings.

Secure Booking & Scheduling – Employers can book workers for specific tasks or durations (hourly, daily, weekly, or monthly).

Payment Integration – Support for cashless transactions via external payment gateways.

Location-Based Matching – Users can find nearby Trabahadors using GPS-enabled search.

Messaging & Notifications – Secure in-app communication for job discussions and real-time updates.

User Verification & Safety – ID verification and feedback system to ensure platform security and credibility.

Limitations and Restrictions

Users Cannot Cancel Bookings/Postings Once Accepted – Once a worker has accepted a booking or posting, it cannot be canceled.

Single Active Job Limit – Users can book only one job at a time, ensuring focused management of each job.

Single Active Booking for Workers – Workers can accept only one job at a time to ensure quality service.

What the Software Will Not Do

Full-Time Job Recruitment – TaraTrabaho is not designed for long-term corporate employment or high-skilled job placements.

Payroll & Benefits Management – The platform does not handle employer-employee benefits, contracts, or government labor compliance.

Guarantee of Work – While it provides a platform for connections, it does not guarantee job availability or worker performance.

Application & Benefits

- **For Employers:** Provides a quick and convenient way to hire workers on demand for various household or personal tasks.
- **For Trabahadors:** Offers flexible job opportunities, helping them earn income and gain work experience.
- **For Society:** Helps address unemployment and provides an avenue for teenagers and unskilled workers to develop essential work skills.

1.3. Definitions, Acronyms and Abbreviations

- *provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS*

Trabahador – A term used in the Philippines referring to a worker or laborer. In this app, it refers to individuals offering services for temporary or short-term jobs.

Instant Job Request – A feature that allows users to send job offers directly to available Trabahadors in real-time.

1.4. References

- *Provide a complete list of all documents referenced elsewhere in the SRS;*
- *Identify each document by title, report number (if applicable), date, and publishing organization;*
- *Specify the sources from which the references can be obtained.*

2. Overall Description

Tarabaho: Tara! Trabaho is a platform designed to connect individuals seeking help with day-to-day tasks and those looking for job opportunities or side hustles. Whether it's running errands, cleaning, tutoring, or other tasks, Tarabaho makes it easy to hire or be hired for specific jobs or durations.

The platform empowers Filipinos, especially those seeking flexible work, by providing a gateway to earn, learn, and grow all while fostering a sense of community and collaboration.

For teenagers aged 15-18, Tarabaho places a special emphasis on safety, balance, and responsibility. To ensure their well-being, teens are only permitted to perform tasks suitable for their age group, such as tutoring, light errands, or pet sitting. Additionally, their work hours are strictly limited to a few hours per day to prioritize their education and personal development while giving them the opportunity to learn the value of responsibility and earning.

With built-in tools for safe communication, task tracking, and secure payments, Tarabaho is committed to creating a safe and user-friendly environment for all users. The platform also encourages mentorship opportunities, allowing young workers to gain valuable experience and skills while helping others in their community.

2.1. Product perspective

- *Put software product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.*
- *A block diagram showing the major components of the larger system, interconnections, and external inter- faces can be helpful.*
- *Describe the modular decomposition of the components using the format below:*

Module 1

Transaction 1.1

Transaction 1.2

Module 2

Transaction 2.1

Transaction 2.2

...

TaraTrabaho is a **standalone** web and mobile application designed to facilitate job matching for short-term or on-demand work. The platform enables users to connect with available Trabahadors based on their immediate needs, eliminating the hassle of traditional hiring processes. While the application operates independently, it integrates with **external services** such as:

- **Payment Gateways (e.g., GCash, PayMaya, or PayPal)** – For secure in-app transactions.
- **Mapping Services (e.g., Google Maps API)** – For location-based job matching and navigation.
- **Third-Party Authentication (e.g., Google Sign-In, Facebook Login)** – For seamless user registration and verification.

Modular Decomposition

Module 1: User Management

Handles all operations related to users (Trabahadors & Employers/Users).

- **Transaction 1.1** – User Registration & Login (via email, Google, or Facebook).
- **Transaction 1.2** – Profile Management (Edit user details, work history, ratings).

Module 2: Job Matching System

Connects employers with Trabahadors based on real-time needs.

- **Transaction 2.1** – Instant Job Requests (Employers send direct offers to Trabahadors).

Module 3: Booking & Scheduling

Allows users to set work durations and availability.

- **Transaction 3.1** – Scheduling (Trabahadors set available work hours).
- **Transaction 3.2** – Booking Confirmation (Employers/Users assign jobs).

Module 4: Payment Processing

Ensures secure and seamless payments.

- **Transaction 4.1** – Job Pricing & Cost Estimation.
- **Transaction 4.2** – Payment Transactions (via GCash, PayMaya, Credit/Debit cards).

Module 5: Rating & Review System

Provides feedback to enhance reliability.

- **Transaction 5.1** – Users rate & review Trabahadors.
- **Transaction 5.2** – Rating-based job recommendations.

Module 6: Notification System

Keeps users updated on job offers and status.

- **Transaction 6.1** – Real-time job alerts via push notifications for Trabahadors.
- **Transaction 6.2** – In-app messaging between employers and Trabahadors.

2.2. User characteristics

- *Describe all user types and their roles and privileges in the system*

1. Worker / Trabahador

Role: The "Worker" is someone who performs tasks or services within the platform. This user type is typically involved in hands-on work and may be hired for specific projects or tasks.

Privileges:

- **View Assigned Jobs/Tasks:** The Worker can view their tasks, deadlines, and any project details they are working on.
- **Update Task Status:** They can mark tasks as in-progress or completed.
- **Communicate with Clients/Employers:** They may have a messaging feature to ask for clarifications or report issues.
- **Track Work Hours/Progress:** If time tracking is enabled, the Worker can log hours worked and update their progress on tasks.
- **Request Assistance or Report Problems:** Workers can request help or report issues they face with the task at hand.

2. User / Client

Role: The "User" (or Client) is someone who hires Workers for specific tasks or projects. They can search for Workers based on specific qualifications or skills needed for a job.

Privileges:

- **Search for Workers:** Users can search for Workers based on specific criteria such as skills, experience, or location.
- **View Worker Profiles:** Users can view the profiles of potential Workers, including their qualifications, skills, previous work, and ratings from other Clients.
- **Assign Tasks to Workers:** After selecting a Worker, Users can assign tasks or projects to them.
- **Monitor Task Progress:** Users can track the progress of tasks assigned to Workers.
- **Leave Feedback/Reviews:** After a task is completed, Users can provide feedback or rate the Worker's performance.
- **Manage Payments:** Users can make payments to Workers according to agreed-upon terms (e.g., hourly, milestone-based, or fixed price).

3. Administrator

Role: The "Administrator" is the highest-level user who manages the system, overseeing all activities and ensuring smooth operation of the platform.

Privileges:

- **User Management:** Admins can create, modify, delete, or suspend User, Worker, or Client accounts, and enforce platform policies.
- **View All Active Jobs/Tasks:** Admins have visibility over all tasks or projects posted on the platform, regardless of whether they are assigned.
- **Monitor Payments:** Admins can oversee all financial transactions on the platform, ensuring proper payment processing.
- **Approve or Reject Content:** Admins review and approve User profiles, Worker profiles, job listings, or reviews to ensure compliance with platform guidelines.
- **Analytics and Reporting:** Admins have access to analytics regarding platform usage, financials, Worker performance, and overall system health.
- **Moderate Communication:** Admins may monitor or moderate communications between Users and Workers to ensure appropriate behavior.
- **Manage System Settings:** Admins can adjust system settings, including pricing, payment methods, and user access levels.

2.4. Constraints

- *Provide a general description of any other items that will limit the developer's options.*
- *Regulatory policies;*
- *Hardware limitations (e.g., signal timing requirements);*
- *Interfaces to other applications;*
- *Parallel operation;*
- *Audit functions;*
- *Control functions;*
- *Reliability requirements;*
- *Criticality of the application;*
- *Safety and security considerations.*

1. Regulatory Policies

- **Data Protection Laws:** The platform must comply with regulations like the GDPR (General Data Protection Regulation) or local data privacy laws to ensure user data is collected, stored, and processed lawfully.
- **Labor Laws:** Since the platform involves employment or job postings, it may need to comply with local labor laws regarding working hours, minimum wage, and other employee rights.
- **Financial Regulations:** Since the platform involves payments or financial transactions, it must comply with financial regulations such as those set by the Central Bank or relevant authorities to ensure secure and lawful transactions.

2. Hardware Limitations

- **Server and Network Requirements:** The platform's performance depends on the hardware and server capacity. Limited resources like slow network speeds or inadequate server capacity could lead to delays in processing data or handling concurrent users.
- **Mobile Device Compatibility:** If the platform supports mobile apps, it must consider different hardware specifications and screen sizes for various devices (phones, tablets, etc.).

3. Interfaces to Other Applications

- **Third-Party Integrations:** The platform might need to interface with external applications or services like payment gateways, email systems, or user verification systems (e.g., using social media logins).
- **API Compatibility:** If external systems need to interact with the platform, developers will have to ensure APIs are compatible, secure, and handle communication efficiently without causing conflicts.

4. Parallel Operation

- **Concurrency and Scalability:** The platform must support multiple users, Workers, and Clients interacting with the system simultaneously without performance degradation. Ensuring database concurrency, task assignment, and updates are managed correctly will be crucial.
- **Load Balancing:** If high traffic is expected, parallel processing must be efficiently managed by load balancers to ensure continuous availability and responsive user experience.

5. Audit Functions

- **Logging and Monitoring:** The system needs robust logging and monitoring capabilities to track activities within the platform. This includes user actions, system errors, and financial transactions, which can be crucial for troubleshooting and compliance audits.
- **Audit Trails:** Keeping an audit trail of key actions (such as account creation, task assignment, and payment processing) is important for compliance and accountability, especially for admins and clients.

6. Control Functions

- **Admin Controls:** The platform must offer administrators fine-grained control over user management, task approval, content moderation, and system settings to ensure compliance with rules and efficient operation.
- **Role-Based Access Control (RBAC):** The platform must ensure that users, Workers/Trabahadors, and Admins can only access and perform actions appropriate to their role. This includes defining and restricting access to sensitive data or system functions.

7. Reliability Requirements

- **System Availability:** The platform should ensure high availability, with minimal downtime, especially during peak usage times. Uptime requirements can be defined (e.g., 99.9% uptime).
- **Fault Tolerance:** It should be able to recover from hardware failures or network issues. Data should be backed up regularly, and the system should be able to resume operations smoothly after a failure.
- **Data Integrity:** The system must ensure that the data remains consistent and accurate, even in the event of a system crash or other issues.

8. Criticality of the Application

- **High Availability for Payments:** If the platform includes payment functionality, it is critical that this part of the system remains available at all times to avoid disruption to financial transactions.
- **Communication Reliability:** If communication between Users and Workers is essential for task coordination, the system should prioritize reliable messaging or notification delivery.
- **Performance under Load:** The platform must handle high usage scenarios, particularly during busy times, without compromising performance or service quality.

9. Safety and Security Considerations

- **User Data Security:** Protecting user data (including payment details, personal information, and communications) is critical. Encryption, secure connections (SSL/TLS), and compliance with security standards (e.g., PCI-DSS for payment data) must be implemented.
- **Access Control and Authentication:** Strong authentication mechanisms should be in place to prevent unauthorized access. Multi-factor authentication (MFA) might be necessary for higher-risk operations.
- **Fraud Prevention:** There must be safeguards in place to detect and prevent fraudulent activities, especially in payment processing and account management.
- **Privacy Concerns:** Given the personal nature of user profiles and interactions, privacy policies must be clear, and users must consent to how their data is collected, stored, and shared.

2.5. Assumptions and dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

3. Specific Requirements

3.1. External interface requirements

3.1.1. Hardware interfaces

This should specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (number of ports, instruction sets, etc.). It also covers such matters as what devices are to be supported, how they are to be supported, and protocols. For example, terminal support may specify full-screen support as opposed to line-by-line support.

3.1.2. Software interfaces

This should specify the use of other required software products (e.g., a data management system, an operating system, or a mathematical package), and interfaces with other application systems (e.g., the linkage between an accounts receivable system and a general ledger system).

The job posting application follows a **client-server architecture**, ensuring scalability and seamless communication between different components. The software interfaces required for the system are as follows:

1. **Web and Mobile Interfaces:**

- The system will provide at least one **web-based interface** and one **mobile application** to ensure accessibility across different platforms.
- The **web frontend** will be developed using **ReactJS** or a similar framework.
- The **mobile application** will be built using **Android Kotlin** to support native Android users.

2. **Server-Side Development:**

- The backend server will be developed using **Java Spring Boot**, providing a robust and scalable API-driven architecture for handling business logic and data processing.

3. **Database Integration:**

- The system will use **Supabase**, an external managed database, for secure and efficient data storage. Supabase offers real-time data synchronization and authentication features, making it a suitable choice for both web and mobile applications.

4. **Payment Integration:**

- The application will integrate **PayMongo** as the primary payment API to handle transactions, ensuring secure and efficient processing of payments for job postings and hiring services.

5. **Maps and Location Services:**

- The application will integrate a **Maps API** (e.g., **Google Maps API** or **OpenStreetMap API**) to enable location-based job postings and search functionality.
- Features may include:
 - Displaying job locations on an interactive map.
 - Allowing users to filter jobs based on proximity.
 - Providing directions to job locations within the mobile app.

3.1.3. Communications interfaces

This should specify the various interfaces to communications such as local network protocols, etc.

3.2. Functional requirements

Web Application Function

- **User Profile Customization**
 - Enables both users and workers to personalize their profiles, including adding photos, bios, and skill descriptions.
- **Booking System**
 - Allows users to book workers for specific daily tasks with flexible scheduling options.
- **Bookmarks / Favorites**
 - Lets users save and bookmark their preferred workers for easy access in the future.

Mobile Application Function

- **GPS Tracker**
 - Provides real-time location tracking for users and workers to ensure efficient task coordination and safety.
- **Chat System**
 - Enables seamless communication between users and workers for task discussions and updates.
- **Payment Integration**
 - Supports multiple payment methods for users (e.g., credit cards, e-wallets) and ensures workers can receive payments through various channels.

2.2 Transaction Name

- Use Case Diagram
- Use Case Description
- Activity Diagram
- Wireframe

...

3.4 Non-functional requirements

Performance

[Details](#)

Security

[Details](#)

Reliability

[Details](#)