Hamdard University
Department of Computing
Final Year Project

Digital Chef Sync

Software Requirements Specifications

Submitted by
Anas Sabih Sarwar (2600-2021)

Abu Bakar (2167-2021)

Hamza Rafique (2195-2021)

Supervisor
Sir Iqbal Uddin Khan

Spring 2025

Document Sign off Sheet

Document Information

Project Title	Digital Chef Sync
Project Code	FYP-001/SP25
Document Name	Software Requirements Specifications
Document Version	1.0
Document Identifier	FYP-001/SP25-SRS
Document Status	Final
Authors	Anas Sabih Sarwar, Abu Bakar, Hamza Rafique
Approver	Sir Iqbal Uddin Khan
Issue Date	01-July-2025

Name	Role	Signature	Date
Anas Sabih Sarwar	Team Lead		
Abu Bakar Balouch	Team Member 2		
Hamza Rafique	Team Member 3		
Sir Iqbal Uddin Khan	Supervisor		
Sir Prem Sagar	Project Coordinator		

Revision History

Date	Version	Description	Author
19-June-2025	0.1	First Draft	Anas Sabih , Hamza
26-June-2025	0.2	Revised Version	Sir Iqbal Uddin Khan
01-July-2025	1.0	Final Report	Anas Sabih

Definition of Terms, Acronyms, and Abbreviations

TERM	DESCRIPTION
Scrum	An agile project management methodology emphasizing iterative development and continuous feedback.
UI/UX	User Interface/User Experience design, focused on usability, simplicity, and accessibility for both job seekers and recruiters.
MongoDB	A flexible NoSQL database used for scalable and schema-less data storage in real-time applications.
React Native	A JavaScript framework used to build cross-platform mobile applications for Android with a native feel.
Frontend	The mobile app interface developed using React Native, allowing users to interact with job features, wallet, and notifications.
Backend	Server-side logic and APIs developed using Node.js and Express.js, handling authentication, job data, payments, and more.

Table of Contents

1. Introduction

The objective of this project is to offer a smart and scalable platform to revolutionize the recruitment and workforce management process within the food industry. Traditional hiring practices, especially in restaurants and event management, often rely on manual methods that lead to delays, mismatched staff, and communication gaps. Digital Chef Sync addresses these challenges by introducing automation, real-time job matching, and efficient financial management. By integrating skill-based filtering, in-app wallets, time tracking, and smart scheduling, the platform streamlines operations and improves reliability for both recruiters and job seekers. It ensures secure data handling through access control, provides instant updates, and adapts easily to scaling demands, making it a future-ready solution. The system is developed using the Scrum methodology to ensure continuous improvement and stakeholder involvement throughout the development cycle. With this platform, the hiring process becomes faster, more transparent, and easier to manage — ultimately benefiting both businesses and their workforce.

1.1 Purpose of Document

The purpose of this Software Requirements Specification (SRS) document is to define the complete functional and non-functional requirements of the **Digital Chef Sync** platform. This document acts as a foundation for the development team, detailing the technical needs, user expectations, and system behavior necessary for successful implementation. Developed using the **Scrum methodology**, this SRS ensures that all team members — including developers, testers, project stakeholders, and product owners — have a clear and shared understanding of the system's objectives and scope. It serves as a roadmap to guide development, monitor milestones, and ensure the final product aligns with user needs and business goals.

1.2 Intended Audience

Development Team:
This document is intended for frontend and backend developers, UI/UX designers, and system architects who are responsible for designing, implementing, and maintaining the Digital Chef Sync platform.

Quality Assurance Team:
QA engineers and testers who will validate the platform's functionality, usability, and performance through systematic testing across various modules including job matching, time tracking, and payment processing.

Project Managers:
Individuals managing the Scrum-based development cycle, ensuring timely delivery of sprints, resolving blockers, and aligning the project with business goals.

Business Stakeholders:
Restaurant owners, event staffing managers, and hiring coordinators who will use the platform to recruit staff and streamline workforce operations. This also includes investors or partners interested in the system's business potential and performance.

System Administrators:
Personnel responsible for managing user roles, access permissions, and maintaining the security and uptime of the platform.

Regulatory Bodies (if applicable):
Any governing or compliance authorities ensuring that the system meets labor regulations, payroll transparency, and data security standards.

2. Overall System Description

2.1 Project Background

The hiring and workforce management processes within the food service industry — especially for restaurants and event companies — have long been handled through outdated, manual methods such as phone calls, walk-in applications, and paper-based records. These processes result in inefficiencies, delayed decision-making, and mismatched job placements. As the demand for skilled and on-demand staff increases in hospitality, the limitations of the traditional hiring system have become more evident, including difficulty in filtering candidates by skills, availability, or role-specific needs.

Furthermore, there is often no streamlined method for shift scheduling, secure payment tracking, or real-time communication between job seekers and recruiters. These gaps frequently lead to missed opportunities, scheduling conflicts, and payment mismanagement, affecting both worker satisfaction and operational efficiency.

To address these ongoing challenges, **Digital Chef Sync** introduces a smart, automated platform tailored for the food industry. It provides real-time job-role matching, secure in-app salary payments, integrated time tracking, and instant notifications — all designed to reduce hiring time, increase reliability, and support scaling needs. The system is built to support both recruiters and job seekers, providing a more accurate, faster, and digitally secure hiring process that aligns with modern industry demands.

2.2 Problem Statement

The food service industry faces critical challenges when relying on manual staffing practices. One of the primary issues is the time-consuming and inconsistent method of collecting candidate data. Often, restaurant or event managers gather information through spreadsheets or informal chats, which leads to data inconsistency, duplication, and errors in candidate selection. There is no smart system in place to automatically match candidates based on role, experience, or shift availability, which results in inefficient placements and over- or under-staffing.

Similarly, the lack of automated payment verification systems causes delays in processing salaries, which can lead to trust issues and dissatisfaction among workers. Staff payments are usually tracked manually, with little to no transparency for either party. Shift scheduling is another manual component that results in poor coordination, mismanaged hours, and last-minute cancellations.

Additionally, communication gaps further complicate the workflow. Recruiters often rely on phone calls or messaging apps to contact candidates, making the process fragmented and difficult to manage at scale. Workers do not receive timely updates about job confirmations, payment status, or upcoming shifts, leading to frustration and uncertainty.

Digital Chef Sync is proposed to eliminate these issues by automating job-role matching, communication, time tracking, and financial transactions — bringing everything into one streamlined platform that is scalable, secure, and designed to meet the specific needs of the food service sector.

2.3 Project Scope

The core aim of this project is to automate and simplify the hiring and staff management process for the food service industry. Digital Chef Sync is designed to reduce the heavy reliance on manual hiring practices by offering real-time job-role matching, smart scheduling, and integrated communication. The system focuses on secure and transparent staff selection by using encrypted data handling and role-based access controls. It caters to both recruiters (restaurants, cafes, and event managers) and job seekers (chefs, waiters, kitchen staff), improving the experience on both ends. In addition, it offers in-app salary payments, automated invoicing, and time tracking for greater operational efficiency. The platform is developed to be scalable and adaptable for future industry changes or feature enhancements, making it a long-term solution for staffing needs.

2.4 Not In Scope

This version of Digital Chef Sync does not include manual interventions beyond initial user onboarding or verification. It will not integrate with broader third-party HR systems or external food industry CRMs at this stage. Additionally, functionalities such as AI-based job prediction, resume parsing, or automated performance evaluations are not part of this release. The platform is limited to pre-hiring, hiring, and payroll functionalities — post-hiring activities like long-term employee training, performance reviews, or promotions are beyond the current development scope.

2.5 Project Objectives

· **Reduce Manual Dependency:** Eliminate traditional hiring delays and errors by automating staff shortlisting, scheduling, and communication.

· **Secure Data Handling:** Protect sensitive user and payment data using encryption and access-level control.

· **Operational Efficiency:** Minimize time spent on administrative tasks by integrating invoicing, time tracking, and shift management.

· **User-Centric Design:** Provide an easy-to-use platform where both job seekers and recruiters can interact with minimal training.

· **Real-Time Updates:** Enable instant notifications for job confirmations, salary payments, and shift changes.

· **Scalability:** Ensure the platform can handle large volumes of users and transactions as business needs grow.

· **Transparency:** Allow clear recordkeeping of job activities, communication logs, and financial histories.

· **Compliance-Ready:** Align with digital employment regulations and industry standards where applicable.

2.6 Stakeholders & Affected Groups

Job Seekers (Chefs, Waiters, Kitchen Staff):
These are the primary users of Digital Chef Sync. They benefit directly from fast job-role matching, instant notifications, in-app salary handling, and the ability to apply based on availability and skills.

Recruiters (Restaurants & Event Managers):
Businesses and employers who manage workforce hiring will experience improved efficiency, reduced staffing errors, and better scheduling through the platform's automated tools.

Administrative & HR Personnel:
Staff responsible for overseeing recruitment processes will see reduced manual effort, better time tracking, and easier financial recordkeeping.

Project Managers:
Individuals managing the platform's development and lifecycle, ensuring alignment with project goals, deadlines, and ongoing system improvements.

Business Owners & Investors:
Stakeholders interested in platform performance, scalability, and return on investment. They rely on analytics, uptime, and usability reports.

Security & Compliance Teams:
Responsible for ensuring that sensitive user and financial data is securely handled through encryption, secure access control, and compliance with digital labor standards.

2.7 Operating Environment

Digital Chef Sync is designed to function as a cross-platform system, accessible via mobile devices. The platform supports real-time interactions, making it ideal for recruiters and job seekers in fast-paced food service environments.

The **frontend** of the application is developed using **React Native** for mobile compatibility and responsiveness, ensuring seamless user experience on both Android and iOS devices. For the administrative web portal or dashboard, React is used to maintain a clean and intuitive interface.

The **backend** is built using **Node.js**, handling business logic, API integration, and real-time communication between the database and the application. **MongoDB** serves as the database, offering a flexible, document-based data structure that aligns well with dynamic job profiles, shift records, and transaction history.

The platform is hosted on a reliable server environment that ensures **24/7 accessibility**, **high uptime**, and **secure data processing**. Proper security protocols including **authentication**, **data validation**, and **role-based access control** are implemented to safeguard sensitive information like user profiles, payment history, and scheduling data.

The system is lightweight, modular, and built to scale with increasing users and data volume — supporting the long-term growth of recruitment needs in the food service industry.

2.8 System Constraints

Digital Chef Sync is being developed within the limitations of a final-year academic timeline, which imposes a strict deadline for planning, development, and testing phases. The team also operates under a **limited budget**, restricting the use of paid APIs, advanced cloud hosting services, or high-end infrastructure.

Additionally, since the platform deals with **financial data and staff information**, it must align with basic data protection guidelines and institutional ethics policies. The project team size is small(3 Members), and members have specific skillsets, which may influence the pace and scope of development. Technical constraints also include testing on limited device types and browser environments due to lack of diverse hardware resources.

2.9 Assumptions & Dependencies

The successful deployment and performance of Digital Chef Sync rely on several critical assumptions:

- **Stable Internet Connectivity** for both job seekers and recruiters is assumed, as the system functions in real time.
- **Development Tools & Testing Devices** such as laptops, mobile phones, and basic hosting platforms will remain available throughout the project.
- The project assumes **continued cooperation and feedback** from restaurants, chefs, or recruitment representatives (or simulated users in testing) to gather accurate requirements and conduct validation.
- The platform depends on various **open-source tools and frameworks** such as Visual Studio Code for coding, **MongoDB** for database handling, and **Node.js** for backend operations.
- Lastly, the team assumes a **consistent SCRUM development flow**, with timely sprint reviews and retrospectives to keep the system aligned with evolving requirements.

3. External Interface Requirements

3.1 Hardware Interfaces

To function effectively, Digital Chef Sync requires certain hardware specifications on both the user and development ends. On the **user side**, any device capable of running a modern web browser or supporting mobile applications is sufficient — including desktops, laptops, tablets, and smartphones.

For the **development side**, systems used by the team should meet a **minimum configuration** of:

- Intel Core i5 (or equivalent) processor
- 8GB RAM
- 500GB SSD storage

These specifications are essential to ensure smooth development, testing, and deployment workflows.
For **data backup and redundancy**, external hard drives with at least 250GB storage capacity will be used periodically. Server hosting and deployment will be done on a reliable cloud-based or university-provided environment, ensuring **high availability, scalability, and minimal downtime**.

The system is expected to support seamless data processing, low-latency communication, and consistent access across varying hardware environments.

3.2 Software Interfaces

Digital Chef Sync interfaces with several key software components and tools to deliver its full functionality. These include:

1. Database Interface

- **Name:** MongoDB
- **Owner:** Project Team
- **Interface Details:** The backend system connects with MongoDB using Mongoose (ODM) for structured, scalable, and fast data management. The database stores information such as user profiles, job posts, shift schedules, payment logs, and system logs. Secure API endpoints handle all data transactions.

2. Frontend Framework

- **Name:** React Native
- **Owner:** Project Team
- **Interface Details:** The mobile app is built using React Native to ensure cross-platform compatibility (iOS and Android). It communicates with the backend via RESTful APIs to fetch and update job listings, notifications, user profiles, and schedules in real time.

3. Backend Framework

- **Name:** Node.js with Express.js
- **Owner:** Project Team
- **Interface Details:** The backend handles all core logic such as job matching, wallet transactions, shift management, and user authentication. It interfaces with MongoDB and exposes secure REST APIs for interaction with the frontend.

4. UI/UX Design Tool

- **Name:** Figma
- **Owner:** Third-party provider
- **Interface Details:** Used during the planning and design phase, Figma allows for wireframing and prototyping of all user interfaces. It ensures consistency across mobile and web layouts and helps in rapid design iteration based on feedback.

3.3 Communications Interfaces

Digital Chef Sync depends on multiple communication interfaces to ensure seamless interaction between users, services, and integrated systems. Being a real-time platform for job matching and workforce coordination in the food industry, reliable and secure communication is essential.

The platform operates over the **Internet**, using **HTTPS (Secure HTTP)** protocol to transmit data safely between the frontend (React Native app) and backend (Node.js/Express.js server). This ensures all user credentials, financial transactions, and job-related data remain encrypted and secure during transfer.

Additionally, **push notification services** are integrated to send real-time updates to both job seekers and recruiters. These notifications include application confirmations, shift reminders, payment updates, and scheduling changes. The system may also interface with **SMS gateways** (if extended) to provide additional communication channels.

For secure salary payments and wallet transactions, **RESTful APIs** are used to connect the system with third-party **payment gateways**. These APIs follow industry-standard protocols for secure authentication and transaction logging.

If required in the future, the platform can be extended to communicate with external business systems or HR software via **API integrations**, ensuring adaptability and scalability.

In summary, the system's communication interfaces are designed to be **secure, scalable, and real-time**, supporting its goal of reducing communication gaps and improving efficiency in hospitality recruitment.

4. System Functions / Functional Requirements

4.1 System Functions

Digital Chef Sync is a smart recruitment and workforce management platform tailored for the food industry. It is designed to streamline hiring, shift management, payments, and communication between job seekers (chefs, waiters, staff) and recruiters (restaurants, catering companies).

1. End-User Functions

Job Seekers (Applicants):

- Register an account and create a profile based on skills and availability.
- View and apply for relevant job posts.
- Receive real-time updates on application status, shifts, and payments.
- Access wallet, time-tracking data, and payment history.

Recruiters (Restaurants/Event Managers):

- Post job opportunities and manage listings.
- View applicant profiles filtered by skill, availability, and location.
- Assign shifts, approve time logs, and process payments via the platform.
- Communicate with staff directly through in-app messaging.

2. Operator/Admin Functions

- Manage user accounts, access levels, and role permissions for both job seekers and recruiters.
- Approve or block suspicious activities or accounts.
- Configure system settings such as notification templates, working hours, and fee structures.
- Monitor logs, review performance metrics, and generate detailed reports on hiring activity, payments, and system usage.

3. Support Functions

- Provide technical and onboarding support for new users.
- Address login issues, wallet queries, and payment failures.

- Assist in shift-related changes and application discrepancies.
- Maintain smooth communication between applicants and recruiters.

4. Integration Functions

- Integrate securely with **payment gateways** to facilitate instant, trackable salary transfers.
- Sync with **email and push notification services** for timely alerts.
- Utilize **MongoDB** for structured data storage and **Firebase** (if needed) for lightweight real-time features.
- Provide **REST APIs** to support potential third-party integrations in the future (like HR or accounting software).

System Function Table

Ref #	*Functions*	*Category*	*Attribute*	*Details & Boundary Constraints*
R1.1	Display Job Listings	*Evident*	Interface availability	Ensure active job listings are visible to all verified job seekers on the homepage.
R1.2	Register New Users	Evident	Input validation	Validate fields like name, phone, email, CNIC, and role (Job Seeker or Recruiter).
R1.3	Smart Filtering by Skills & Availability	Evident	Dynamic filter system	Enable recruiters to filter candidates based on selected skills, availability, and location.
R1.4	Store User Profiles in Database	Hidden	Data persistence	All profile data must be securely stored in MongoDB without data loss or duplication.
R1.5	Allow Profile Editing	Evident	Editable fields	Users can update details like working hours, skills,

				address, and contact info.
R1.6	Manage Concurrent Session Handling	Hidden	Load balancing	Platform should support multiple logins and real-time operations without lag or failure.
R1.7	Password Setup & Recovery	Evident	Password policy	Enforce strong passwords and provide recovery via email or OTP-based reset.
R1.8	Generate Unique User ID	Hidden	Unique identifier	System must auto-generate a unique ID for every user upon successful registration.
R1.9	Create Job Posts	Evident	Form fields	Recruiters can specify job title, required skills, shift timing, location, and pay rate.
R1.10	View & Apply to Job	Evident	Application flow	Job seekers can apply for active jobs and receive a confirmation message.
R1.11	In-App Wallet Functionality	Evident	Payment data	Both user can view balance, withdrawal history, and upcoming payments.
R1.12	Shift Scheduling Panel	Evident	Timeline UI	Recruiters can assign staff to time slots and notify them of their shift schedule.
R1.13	Application Summary & Status Updates	Evident	Review & status tracking	Users can track job application statuses such as pending, hired, or completed.
R1.14	Admin Dashboard Interface	Evident	Admin controls	Admin can view user logs, approve/reject posts,

				manage complaints, and export reports.

System Attributes/ Nonfunctional Requirements

Attribute	Details and Boundary Constraints	Category
Response Time	All core actions like job application, wallet updates, and profile edits must respond within 5 seconds to ensure a smooth user experience.	Mandatory
Concurrent User Load	The platform must be able to handle at least **1,000 active users** (recruiters and chefs) at the same time without slowdowns.	Mandatory
Interface Design	The system interface must be browser-based with a **user-friendly, responsive design**, compatible across desktop and mobile.	Mandatory
Accessibility	The platform should follow basic **web accessibility standards (WCAG)** to ensure usability for users with visual or physical impairments.	Mandatory
Scalability	The architecture must allow the system to **expand easily** in the future as more job seekers and companies join the platform.	Optional

4.2 Use Cases

4.2.1 List of Actors

· **Chef (Applicant):**

Registers on the platform, applies for jobs, manages wallet and availability, and tracks job updates.

· **Recruiter (Restaurant/Event Manager):**

Posts jobs, filters candidates based on skills and shift timings, sends hiring offers, and reviews chef profiles.

· **Admin:**

Manages overall system operations including user roles, data monitoring, security updates, and platform integrity.

· **Support Team:**

Provides technical help and user support regarding application issues, wallet concerns, or job posting errors.

· **Payment Gateway:**

Handles secure transactions, wallet top-ups, job payments, and automated invoice verification.

4.2.2 List of Use Cases

· **Register and Apply for Jobs:**

A chef creates a profile, adds skill details, and applies for available jobs based on filters such as shift, location, and skill tags.

· **Track Job and Wallet Status:**

Chefs can view current application status, job history, and wallet balance updates in real-time.

· **Post Job and Filter Candidates:**

Recruiters post a new job with requirements and use advanced filters to shortlist the most relevant chefs based on availability and skill.

· **Send Job Offers and Confirmations:**

Recruiters send real-time offers to selected chefs, who can accept or reject with a single click.

· **Automate Payments and Invoicing:**

Once the job is completed, the system automatically triggers invoice generation and processes payment to the chef's in-app wallet.
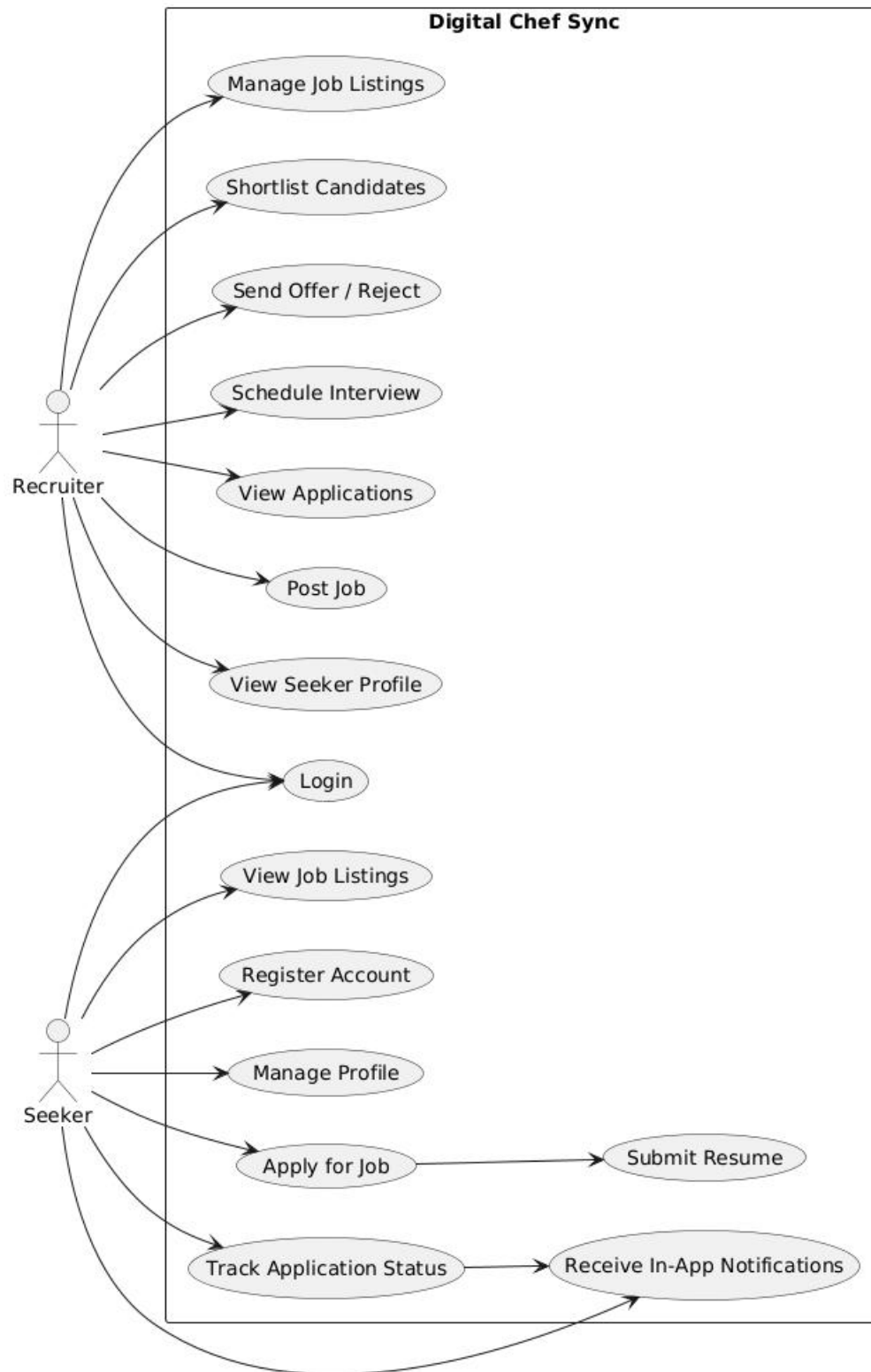
· **Generate Operational Reports:**

Admin can view, export, and manage detailed reports including job completions, wallet transactions, and hiring metrics.

· **Manage User Roles and Access:**

The Admin or IT Team can assign roles (Chef, Recruiter, Staff), manage permission levels, and monitor access logs for security.

4.2.3 Use Case Diagram

4.2.4 Description of Use Cases

Document each use case. This can completed using the tables provided below:

Section: Main

Name:	*Post Job Shift*
Actors:	*Employer (Restaurant / Event Company Admin)*
Purpose:	*Allow an recruiter to create and post a new job shift for chefs, waiters, or other service staff.*
Description:	*An employer logs into the system, fills in shift details such as role, date, time, location, and payment, and posts the shift. The system validates the data, notifies matching candidates, and displays the shift on the job board in real time.*
Cross References:	*Functions: R1.1, R2.3*
	Use Cases: Recruiter must be registered and logged in before posting a shift.
Pre-Conditions	*The system is live and accessible.*
	The recruiter is logged into a verified account.
Successful Post-Conditions	*The shift is saved and listed for relevant candidates.*
	Notifications are sent to qualified users.
Failure Post-Conditions	*The shift is not saved.*
	The system displays errors for invalid/missing fields.

Typical Course of Events			
Actor Action		**System Response**	
1	*Recruiter logs into their account.*	2	*System verifies credentials and grants access*
3	*Recruiter opens "Post a Shift" form.*	4	*System loads the form with fields like role, time, and date.*
5	*Recruiter fills out shift details.*	6	*System validates fields (e.g., date/time format, payment).*
7	*Recruiter submits the shift.*		
		8	*System posts the job and sends notifications to seekers.*

Alternative Course

Step 5:   *Missing or invalid input, system highlights errors and asks for corrections.*

Step 7:   *Network/server error, system shows error message and retries submission.*

Section: Apply for Job Shift

Typical Course of Events

Actor Action		System Response	
1	Seeker logs into the app.	2	System verifies users and grants access.
3	Seeker search available shifts.	4	System filters shifts based on seeker skills and availability.
5	Seeker taps "Apply" on a shift.	6	System records application and updates employer's dashboard.

Alternative Courses

Step 3:   *If seeker account is inactive → prompt to verify profile.*

Step 5:   *Seeker has time conflict → system blocks application*

Section: Payment & Wallet Transaction

Typical Course of Events

Actor Action   *System Response*

1.   *Recruiter selects "Pay Worker" option.*   2. System redirects to wallet interface.

3.   *Recruiter enters payment amount and confirms.*   4. System verifies balance and processes payment.

5.   *Payment is completed.*   6. Receipt is sent to both parties and wallet balance is updated.

Alternative Courses

Step 4:   *Insufficient balance → system prompts top-up or alternate method.*

Step 5:   *Server/payment gateway error → system generates a retry option and logs issue.*

5. Non - Functional Requirements

5.1 Performance Requirements

- The Digital Chef Sync platform must be capable of supporting up to **1,000 chefs and recruiters concurrently** without slowing down.
- All key operations — such as **job applications, shift scheduling, wallet transactions, and status updates** — should be processed within **5 seconds or less**.
- The backend database must efficiently store and retrieve **large volumes of chef profiles, shift logs, and transaction records** without any delay in query results.
- A **step-by-step user guide** should be provided to help chefs and employers with account registration, job application, payment workflows, and shift management.
- **Admin guides** must explain how to manage users, monitor transactions, review job matches, and troubleshoot routine issues.
- The platform should also feature an **FAQs section** and **tutorial videos** to help users resolve common queries on their own without needing direct support.

5.2 Safety Requirements

- The system must be designed to **prevent data loss** during important actions like shift application submission or wallet-based payments by implementing **automatic fail-safe mechanisms**.
- Regular **automated backups** of system data (applications, profiles, transactions) must be performed to prevent data loss from system crashes, network failures, or power outages.
- The system should **log all critical errors**, and notify the administrator team immediately if any major failure or risk is detected to enable rapid resolution.

5.3 Security Requirements

- All communication between the user (whether a chef or recruiter) and the system must be **encrypted using SSL/TLS protocols** to ensure data privacy and prevent unauthorized access.
- The system must enforce **secure login practices**, including hashed passwords, password strength validation, and secure session management.
- **Role-Based Access Control (RBAC)** must be applied so that each user type (chef, employer, admin) only has access to the data and features specific to their role.

- The system should be capable of **detecting unauthorized access attempts**, such as brute-force login attempts, and must temporarily lock accounts or alert administrators when such attempts occur.

5.4 Reliability Requirements

- The **Digital Chef Sync** platform must ensure a **99.9% uptime** during active hiring and staffing cycles to guarantee availability for both chefs and employers.
- Critical operations — including **payment verification, wallet transactions, and job application submissions** — must function without interruption, even during peak usage times like event seasons or weekends.
- The system must be equipped to **handle unexpected failures gracefully**, ensuring no data loss or disruptions to live job requests or shift tracking.

5.5 Usability Requirements

- The user interface must be **simple, user-friendly, and responsive**, ensuring that both chefs and recruiters can navigate the platform with minimal guidance or training.
- The system must meet recognized **web accessibility standards** (e.g., WCAG), making it inclusive for users with visual, motor, or cognitive impairments.
- All forms and pages must include **real-time validations, clear error messages, and helpful prompts** to guide users step-by-step through each task (e.g., applying for a job or verifying payment).

5.6 Supportability Requirements

- The system should be designed to support **smooth updates and patches** without logging out active users or interrupting ongoing sessions.
- Its **modular structure** should allow easy expansion — for example, future integration with external HR tools, SMS services, or advanced analytics features.
- **Comprehensive logs** must be maintained to support quick issue diagnosis and regular maintenance by the IT support team.

5.7 User Documentation

- A **complete user manual** must be provided to chefs and employers, covering how to create accounts, apply for jobs, set availability, make payments, and view wallet balances.

- **Admin documentation** should explain how to assign roles, verify staff credentials, monitor activity, and resolve technical issues.
- An **FAQs section**, **step-by-step tutorials**, and **short video guides** should be available on the platform to help users troubleshoot common problems independently — without needing to contact support.

6. References

Workstream Blog
Available at: https://www.workstream.us/blog/restaurant-hiring-software

QSR Magazine
Available at: https://www.qsrmagazine.com/emerging-tech/how-tech-changing-restaurant-staffing

ToastTab – On The Line
Available at: https://pos.toasttab.com/blog/on-the-line/hiring-restaurant-staff