Software Analysis Document   
Course Project: CS244 Spring 2023

Project Name

Orphan Charity Organization

Team:

Mahmoud Ahmed

Yahya Salah Abdelhameed

Ali Mohamed  
  
Abdelrahman Nabil

Kirolos Refaat

# Introduction

1. The purpose of the document for an orphan charity organization system project is to provide a clear plan that will help the organization to develop a software system to manage their operations and help the children they support. This plan should include important details such as the features that the system will need to have, what kind of technology will be used, and how long it will take to complete the project. It will also provide a budget for the project, who will be responsible for different tasks, and what risks or problems might come up during the process. Overall, the document is important because it will help the organization to make sure that the software system is developed successfully, on time, and will meet the needs of the children they support.
2. The document will provide a detailed overview of the orphan charity organization system project, including its objectives, key features, functionalities, and technical specifications. It will also outline the project's timeline, budget, and resource requirements, as well as any potential risks or challenges that may arise during the development process.
3. The intended audience for the document is the team and stakeholders involved in the orphan charity organization system project. This includes the management team, project managers, developers, and any other relevant personnel.

# Project Drivers

## Purpose of the Project

The purpose of this project is to develop a software system to help manage operations and support the children under the care of an orphan charity organization. The system will allow the organization to organize information about the children, their sponsors, donations, and other key details. It will help automate and streamline many of the organization's processes to allow staff to focus on directly helping the children. The overall goal is to improve the level of care and support the organization can provide to the orphans they serve.

## Stakeholders

This section describes the stakeholders—the people who have an interest in the product.

### The client /Customer

The client for this project is the orphan charity organization itself. As the sponsor of the project, the organization will make the final decision on whether the delivered system meets their needs. Satisfaction of the key stakeholders in the organization is essential to the success of the project.

### The Users

The primary users of the system will be: Organization staff: Mostly non-technical, wide range of ages. They need an easy to use system to enter and access data about the children, sponsors, donations, etc. They are the key users of the system.

Organization management: Also non-technical, responsible for overseeing organization operations and staff. They need access to reports and metrics to monitor the impact and effectiveness of programs. They are secondary users.

Sponsors: Provide financial support to individual children. Mostly non-technical adults. They need a way to submit payments, receive updates on the child they sponsor and have a portal to communicate with organization staff. They are a secondary category of users.

# Project Constraints

-Budget: As a non-profit organization, there is not funding . The final system design must be cost-effective to develop and maintain .

- Technology: The system must utilize cost-effective, user-friendly technology that can be readily learned and used by non-technical staff and stakeholders. Advanced or complex technologies are not suitable given users' needs and capabilities.

- Data/Security: The system will contain sensitive data about children, sponsors, donations, and organization finances. Stringent security standards must be adhered to in order to protect data and privacy. No compromises can be made on security.

- Scalability: The system may need to scale up over time if the organization expands the number of children or programs they support. The final design should allow for moderate future scalability without major redesign.

## Solution Constraints

- The system must be built using PHP and phpMyAdmin, in addition to other open-source or low-cost technologies, given the limited budget. Proprietary or enterprise systems are not viable options.

The preferred technology stack is: › Front-end: HTML, CSS, JavaScript › Back-end: PHP › Database: MySQL (managed via phpMyAdmin)

- For security, the system must implement role-based access control, secure password storage using strong hashing algorithms compatible with PHP, encrypted connections using SSL, SQL injection prevention, and other standard authentication mechanisms. No sensitive data may be stored or transmitted in cleartext.

- The system must be designed for maintainability with well-documented PHP code, separation of concerns (front-end vs back-end logic), and standardization of environments, tools, and processes. Easy maintainability will minimize costs for the organization going forward.

- phpMyAdmin will be used to directly manage the MySQL database for simpler deployment and maintenance needs of non-technical organization staff. Security controls must be put in place to prevent unauthorized access via phpMyAdmin.

## Problem Statement

An orphan charity organization is facing difficulty in efficiently managing their operations due to manual processes, lack of coordination and collaboration among staff, and inadequate tracking of donations and expenses. The organization is looking for a solution that can automate their processes, improve communication among staff members. The organization needs a software system that can help them with the following:

Donor Management, Fundraising Management, Volunteer Management, Expense Management, Collaboration and Communication, Reporting and Analytics through visual dashboards and reports

## Current Situation

The orphan charity organization currently manages most operations and processes manually using a combination of paperwork, in-house databases, and spreadsheets. Some of the major issues with the current situation include:

- Information exists in silos across different staff members and physical locations, making it difficult to get a unified view of key data. There is significant reliance on institutional knowledge.

- Simple tasks like tracking donations, sponsors, and children's' information require a large amount of manual effort and duplication. Much staff time is spent on administrative work rather than directly supporting the children.

- Data security and privacy are a concern given the reliance on spreadsheets and paper documents as well as limited access controls. There is a risk of data loss or unauthorized access.

- Donors and sponsors lack a simple, transparent way to submit funds, communicate with the organization and receive updates on the child they support. The manual processes do not serve them well.

## User Requirements

The orphan charity organization system must meet the following key requirements of its users:

- Provide a unified, integrated view of all data related to organization operations including information on children, sponsors, donations, finances, programs, staff, etc. This will enable transparent reporting and decision making.

- Allow for simple, streamlined entry and maintenance of all child, sponsor, donation and other key records to minimize administrative effort and ensure data accuracy. Minimal duplication of data entry should be necessary.

- Incorporate a permission-based access control system to provide access to sensitive data based on the user's role. Strict security controls must protect privacy and prevent unauthorized data access or modification.

- Provide tools to properly onboard new staff by allowing easy viewing of organizational knowledge embedded in the system. New hires should become quickly acquainted with organization data and processes through the software.

-Use technology that is easy for non-technical staff to learn and utilize with minimal training required. The user interface and experience should be simple, intuitive and consistent across modules or dashboards.

### Business Processes ( use cases)

Identify the processes that are critical to the organization's business objectives. This involves understanding how the organization operates, what tasks are performed, and what resources are required.

### Business Process 1

### Business Process 2

### Business Process n

### Business Rules (Constraints)

These are business rules that might have an impact on the work/business/domain that is the source of the requirements. Relevant business rules will be the trigger for requirements.

### Business Rule 1

Child privacy and consent: No details on a child's personal history or circumstance may be disclosed publicly without obtaining proper consent from parents or legal guardians. All child data in the system must be kept strictly confidential.

### Business Rule 2

Financial audits: As a registered non-profit, the organization must maintain thorough financial records and submit to independent audits of accounts on an annual basis. The system must provide full support for managing budgets, expenditures, donations, and reporting to facilitate audits.

### Business Rule 3

Background checks: All staff and volunteers who interact directly with children must undergo comprehensive background and reference checks before being granted access. Clearance levels should be recorded in the system, and access controls put in place based on an individual's role and clearance status.

### Business Rule 4

Donor recognition: In accordance with established gift acceptance policies, donor recognition and naming opportunities must be provided based on contribution levels. The system should track donation amounts and history to determine appropriate recognition for each donor. Recognition and impacts should be reported back to donors regularly to encourage ongoing support.

## The Scope of the System (Use case diagram)

## Use Case Scenarios:

### Use case 1 scenario

|  |  |
| --- | --- |
| 1. Identifier and name | Admin add user |
| Initiator | User (may be receptionist or accountant or warehouse manager) |
| Goal | Add the user to the system |
| Precondition | None |
| Postcondition | The user will be added to the system and has id, pw and email |
| Assumptions | The expected initiator is a receptionist.  He is not already known to the software system |
| Main success scenario | |
| 1 | The admin select system users from the dashboard to add user |
| 2 | The admin enter user (id, username, email, password) |
| 3 | The admin enters add user. |
| 4 | The user has been added to the system |
| 5 | The system adds the user to the systems database |
| Extensions | |
| 3.a | user already exists |
| 3.a.1 | Database is completed |

### Use case 2 scenario

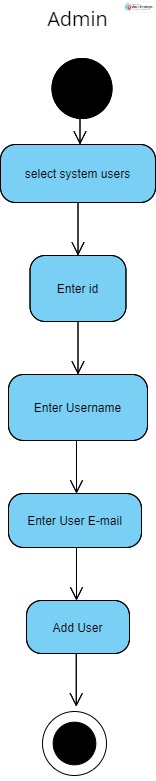
|  |  |
| --- | --- |
| 1. Identifier and name | A receptionist books a room |
| Initiator | The guest (may be receptionist or accountant or warehouse manager or orphan) |
| Goal | A room is reserved for the orphan |
| Precondition | None |
| Postcondition | A room of the desired type will have been reserved for the guest for the requested period. and the room will no longer be free for that period. |
| Assumptions | The expected initiator is a orphan.  He is not already known to the software system |
| Main success scenario | |
| 1 | The receptionist selects book room from the dashboard to books a room |
| 2 | The receptionist enter room (number, orphan name, orphan id, date, time) |
| 3 | The receptionist book room |
| 4 | The system checks the availability of the rooms |
| 5 | The system makes a booking and gives it a number if a room is available |
| 6 | The system cancels a booking and send message “There are no rooms available” if the room is not available. |
| Extensions | |
| 3.a | Room not available |
| 3.a.1 | The receptionist offer another date and time to the orphan |

### Use case 3 scenario

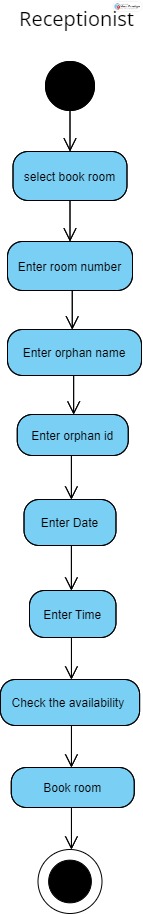
|  |  |
| --- | --- |
| 1. Identifier and name | Accountant provide transactions |
| Initiator | The accountant |
| Goal | Provide the transactions |
| Precondition | None |
| Postcondition | Transactions are provided by the accountant. |
| Assumptions | The expected initiator is a accountant.  He is not already known to the software system |
| Main success scenario | |
| 1 | The accountant selects provide transactions from the dashboard |
| 2 | The accountant enter id and select transaction type and enter the value, date and doner name |
| 3 | The accountant provides transactions. |
| Extensions | |
| 3.a | The donor does not want to be named |
| 3.a.1 | The id of the transactions was missing |

## Activity Diagrams:

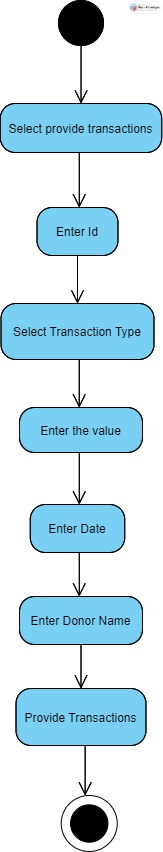
### Use case 1 Activity Diagram



### Use case 2 Activity Diagram



### Use case 3 Activity Diagram



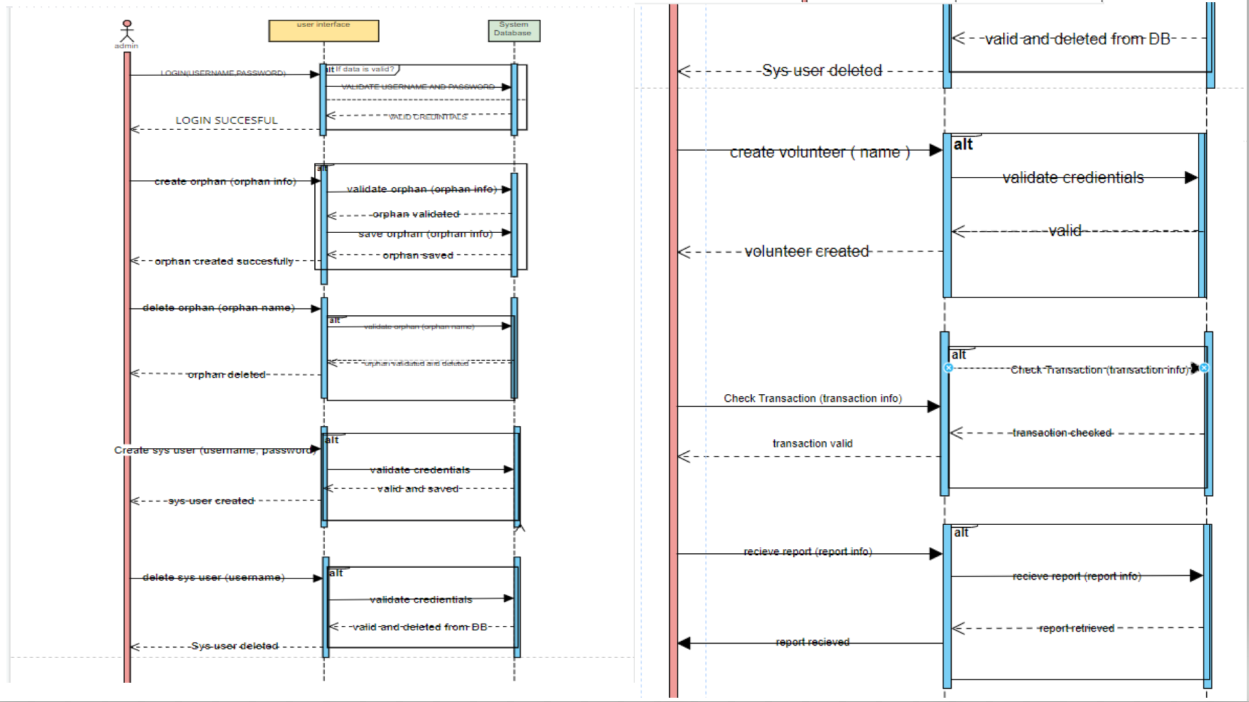
## Class Diagram:

This section should provide a detailed description of the software system's object model, including the classes, attributes, methods, and relationships.

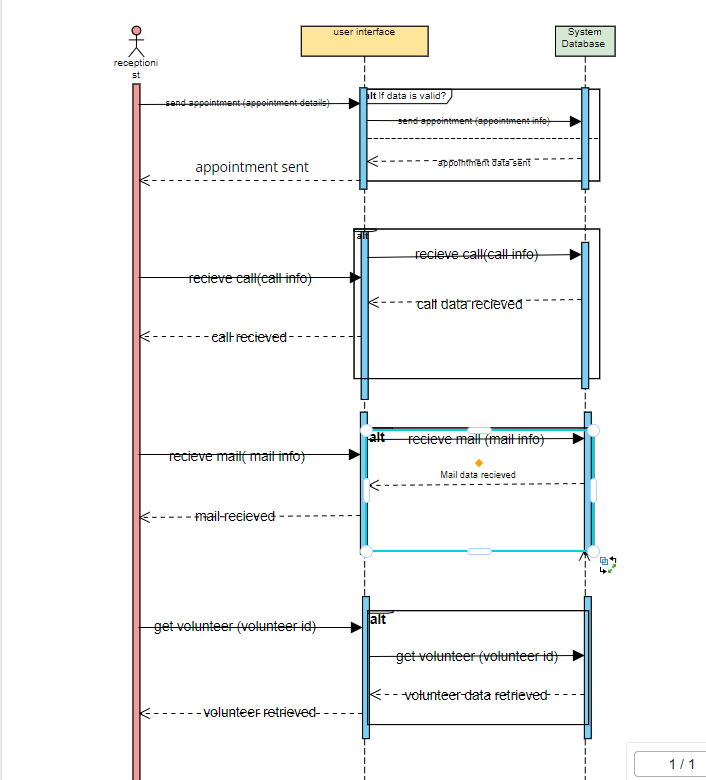
## Sequence Diagrams:

This section should provide a graphical representation of the interactions between the software system's objects and the messages that they exchange.

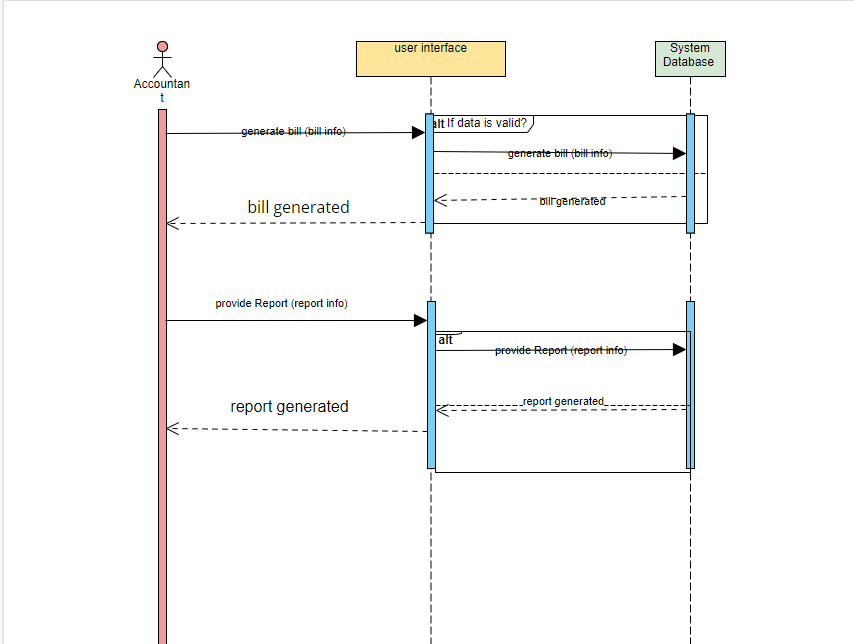
### Use case 1 Sequence Diagram



### Use case 2 Sequence Diagram



### Use case 3 Sequence Diagram



## State Diagrams:

This section should provide a graphical representation of the states and transitions that the software system's objects can undergo.

## Data Model:

The primary data entities in the system include:

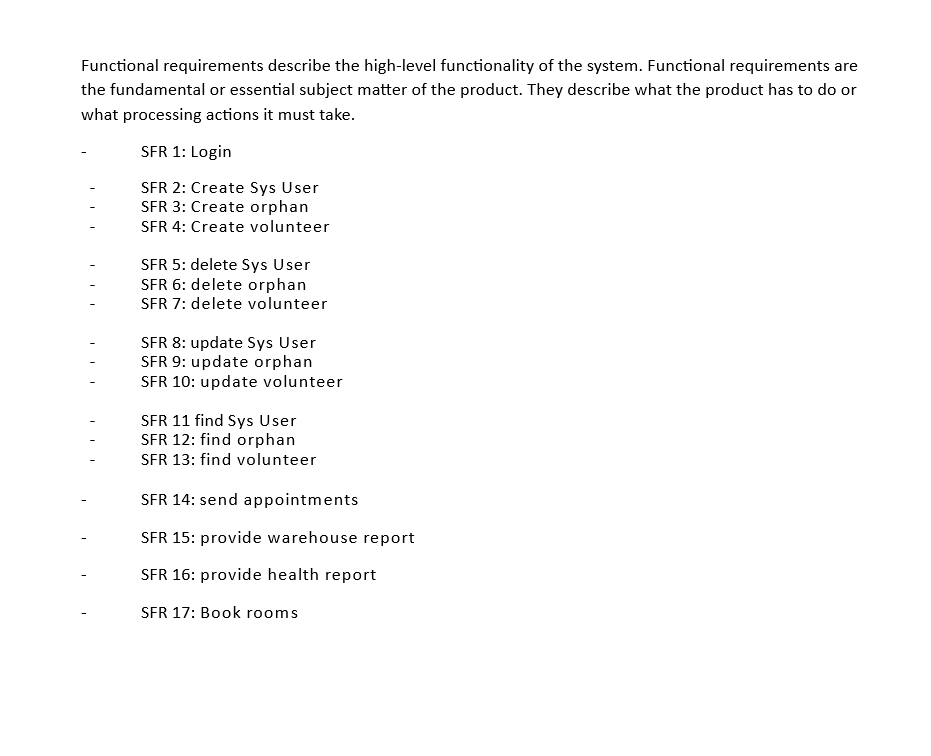
-Children: Name, DOB, gender.

- Donations: Amount, date, payment method.

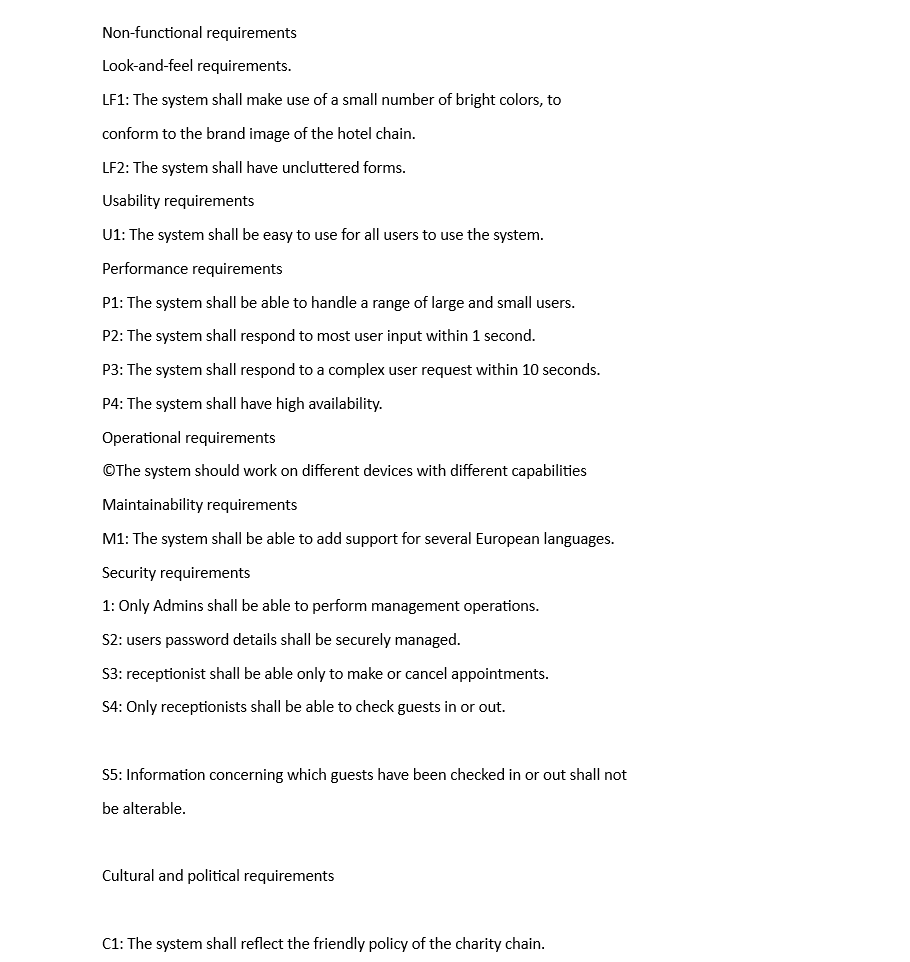
- Volunteers: Contact details.

- Staff: Name, contact details, position ,hire date.

# System Functional Requirements



# System Non-functional Requirements



## Look and Feel Requirements

The color scheme should be muted and professional with high contrast elements for readability. Bright or neon colors should be avoided. A color palette of navy, grey and orange is recommended based on the organization's branding.

Interactive elements like buttons, links and icons should be large, clearly and consistently labeled, and easily distinguished from other text or content. This will reduce confusion and improve usability, especially for older or less technically-inclined users.

System messages or alerts should be displayed prominently at the top of the screen using simple language without technical jargon. They should be highlighted to draw the user's attention.

Data displayed on screen should be logically organized and easy to scan using headings, indentation or other structural elements. Long text passages should be avoided where possible. Bulleted lists, tables and charts/graphs can make information intuitive to view and understand at a glance.

## Usability and Humanity Requirements

## Performance Requirements

To ensure adequate speed, reliability and scalability, the following performance requirements apply:

The system must support a minimum of 50 concurrent staff users, 100 concurrent sponsor/donors, and 200 total users at peak periods (first of month, end-of-year campaign) with average response times of 3 seconds or less for all page loads and transactions. Performance must not significantly degrade beyond these base levels during peak usage times.

The database should be able to store up to 20,000 child profiles, 30,000 sponsor/donor profiles, and 50,000 donation records while queries continue to return results in under 5 seconds for standard reports and administration functions. Archived or less frequently accessed records may take longer to retrieve but still within acceptable limits.

All pages, forms, and user interfaces should load, submit, navigate and refresh quickly without perceived waiting times for users. Delays of more than 3-5 seconds will result in frustration and reduced satisfaction.

## Operational and Environmental Requirements

## Maintainability and Support Requirements

To ensure the system remains operational and sustainable over time, the following maintainability and support requirements apply:

The system must be thoroughly documented including architecture diagrams, data models, configuration settings, installation/upgrade procedures, troubleshooting steps, and all other details required for ongoing maintenance, support, and eventual transition to new technologies. Documentation should be accessible for authorized technical staff.

Code must be well-commented, follow standard conventions, and organized logically for easy maintainability. A naming convention and best practices should be established for consistency. Separation of concerns (UI, business logic, data access) must be enforced in the system design.

Administrative interfaces for managing configurations, user roles, content updates or other changes must be intuitive and well-designed to minimize demands on technical support resources. Options for self-service where possible and UI designs promoting easy understanding and usage should be prioritized.

## Security Requirements

The following security requirements must be implemented to protect the confidentiality, integrity, and availability of data and the system:

All sensitive user data including names, contact details, financial information, and child/case records must be encrypted at rest and in transit using strong encryption standards. No personally identifiable information may be stored or transmitted in cleartext.

User access to data and system functions must be governed by role-based access controls with the principle of least privilege followed. Default restrictive permissions should be put in place with access granted specifically where needed. Staff will have access only to data required for their role.

## Cultural Requirements

## Legal Requirements

# Appendixes

## Important Definitions

***Functional requirements*** are the fundamental or essential subject matter of the

product. They describe what the product has to do or what processing actions it must take.

***Non-functional requirements*** are the properties that the functions must have, such as performance and usability. Do not be deterred by the unfortunate name for this kind of requirements, they are as important as the functional requirements for the product’s success.

***Project constraints*** are restrictions on the product due to the budget or the time available to build the product.

***Design constraints*** impose restrictions on how the product must be designed. For

example, it might have to be implemented in the hand-held device being given to

major customers, or it might have to use the existing servers and desktop computers, or any other hardware, software, or business practice.

***Project drivers*** are the business-related forces. For example, the purpose of the project is a project driver, as are all of the stakeholders—each for different reasons.

***Project issues*** define the conditions under which the project will be done. Our reason for including them as part of the requirements is to present a coherent picture of all factors that contribute to the success or failure of the project and to illustrate how managers can use requirements as input when managing a project.