[Project Name]

Kern County Department of Human Services

Analysis and Requirements

[Analyst Name]

YYYY

# History

This section tracks the history of the document and its various versions.

|  |  |  |
| --- | --- | --- |
| Version | Reason for Change | Updated By |
| 1.0 |  |  |
| 2.0 |  |  |

# Project Summary

*This section should provide a brief, high-level overview of the project*

# Problem Statement

This section should examine the problems that the stakeholders are trying to address with this project. This is usually about a paragraph.

# Business Objectives

This section should examine the **why** of the project. Examine how the project will benefit the company as opposed to what they want the solution to do. Typically, this involves regulatory compliance, cost savings, reduced workload, increased efficiency, less waste, or improved morale/public image.

|  |  |
| --- | --- |
| ID | Description |
| 1.0 | List your business requirements here |
| 2.0 |  |

# Key Performance Indicators

*This section should provide concrete metrics by which we can evaluate how well we have achieved our business objectives (e.g. reduce lobby wait time by 5 minutes). We should be able to draw a connection between a performance metric and a business objective.*

|  |  |
| --- | --- |
| Performance Metric | Business Objective |
|  |  |

# Deliverables

*This section should examine what the project will ultimately produce. Deliverables should be results or output from the project. Things we are creating. Deliverables are not existing infrastructure items or services that will be implemented into other deliverables.*

* Requirements Document
* New Database[[1]](#footnote-2)
* Mobile-friendly, web-based user interface

# Milestones

This section should examine the major accomplishments in your project that will help you monitor your progress. This section does not establish dates, as it will be too early in the project to begin the scheduling process.

|  |  |  |
| --- | --- | --- |
| ID | Milestone | Due Date |
| 1 | Create new database |  |
| 2 |  |  |

# High Level Assumptions

This section should examine high-level assumptions that are made about the project or resources (e.g. “John will be able to work on the project full time” or “we will be able to grant the public access to the app”)

* Assumption 1 (e.g. developer will be able to devote 30 hours/week to this project)]
* Assumption 2
* Etc.

# High Level Risks

This section should examine high-level risks. This includes how severe the risks are and what steps will be taken to mitigate approach will be taken if any. See table for example

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Severity (1-10) | Likelihood (1-10) | Mitigation Plan |
| Users will reject self-service options | 10 | 3 | Survey users to determine how receptive they are to self-service |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Prerequisites

Outline any aspects that need to be in place prior to the design, development, and implementation of the project proposed in this document. Be sure to be clear and concise for all listed prerequisites. Also, clearly outline why each prerequisite is needed.

Note: If no prerequisites are needed, include a paragraph justifying why there are no prerequisites.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Prerequisite | Description | Completion Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Business Requirements

This section should examine **what** the proposed solution is supposed to do. In more depth than the business objects, what does the business want to accomplish. This section should not examine the **how**. If you start talking about technical matters (e.g. send a text message), you have strayed off course.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Business Objective ID | Title | Description |
| 1.0 | 1.0 | Write a Short Title for the Requirement (e.g. Customer Appointment Reminder) | Write a more detailed requirement for the requirement (e.g. Send the customer a reminder for the appointment 24 hours before the appointment date/time) |
|  |  |  |  |

# Functional Requirements

This section should examine **what** the project deliverables will do (e.g. send daily summary emails to users). These are not use cases, so you do not need to be overly detailed. Something as simple as “Users should receive text messages 48 hours before their appointment (within a 10-minute window) that will include the date/time of the appointment. If the user replies ’Yes’, the appointment is confirmed. If they reply ‘No’, the appointment is cancelled. No reply will leave the appointment in the ‘Pending Confirmation’ status.” While this spells out how the feature should work, it leaves implementation details to the developer.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Business Requirement Mapping(s) | Title | Description |
| 1.0 |  |  |  |
| 1.1 |  |  |  |
| 2.0 |  |  |  |
| 3.0 |  |  |  |

# Non-Functional Requirements

This section should examine requirements that do not provide functionality but that are still necessary for the success of the project (e.g. API should follow REST design, application should be written using Vue or ASP.NET, emails must be sent within 2 minutes of clicking the submit button, or must be IE 11 compatible)

|  |  |  |
| --- | --- | --- |
| ID | Title | Description |
| 1.0 |  |  |
| 1.1 |  |  |
| 1.2 |  |  |
| 2.0 |  |  |
| 2.1 |  |  |

## Performance Requirements

This section examines non-functional requirements specific to performance (e.g. app must load within 3 seconds on average or notification must be sent within 2 minutes of submitting form). If no performance requirements are specified, delete this section, or simply put N/A

N/A

## Availability Requirements

This section examines non-functional requirements specific to availability (e.g. app should be available between the hours or 8 AM and 5 PM). If no availability requirements are specified, delete this section, or simply put N/A.

N/A

## Capacity Requirements

This section examines non-functional requirements specific to capacity (e.g., the SMS API needs to support 800 text messages in a 2-minute window).

N/A

## Security & Authorization Requirements

This section examines who can access what for one or more deliverables as well as the security of the application and its data in general (e.g., the password field must be hashed).

* Security Requirement 1 (e.g. Employees with authenticate using Azure Open ID)
* Security Requirement 2

## Monitoring and Reporting Requirements

This section examines what data needs to be logged, how frequently it needs to be logged, and what kind of reporting needs to be done based on that data.

* Monitoring Requirement 1 (e.g. all user logins must be logged for review)
* Monitoring Requirement 2
* Reporting Requirement 1 (e.g. admins should be able to pull up a report to review after hours logins)

# Success Criteria

This section examines what criteria must be met to call the project a success (e.g. “all KPIs are met” or “case info is updated more frequently (users log in at least 2 times per week)”).

None Identified

# Project Management/Development Methodology

This section examines the selected methodologies for project management and development, the rationale behind the selected methodologies, and how those methodologies will be carried out.

## Selected Methodology

## Justification

# System Design

This section examines the more technical aspects of the project like architecture, class design, and database design.

## Proposed Architecture

This section will examine the proposed architecture for the system and the reasons behind selecting that architecture.

## Components

Provide a list of components (APIs, Systems, or Network Resources) that will be used in the system and the purpose of each components creation/integration into the system.

|  |  |  |
| --- | --- | --- |
| ID | Name | Description/Purpose |
|  |  |  |
|  |  |  |

### Architecture Diagram

UML Component diagram (or other visual representation) of the various components of the system and they will interact with each other

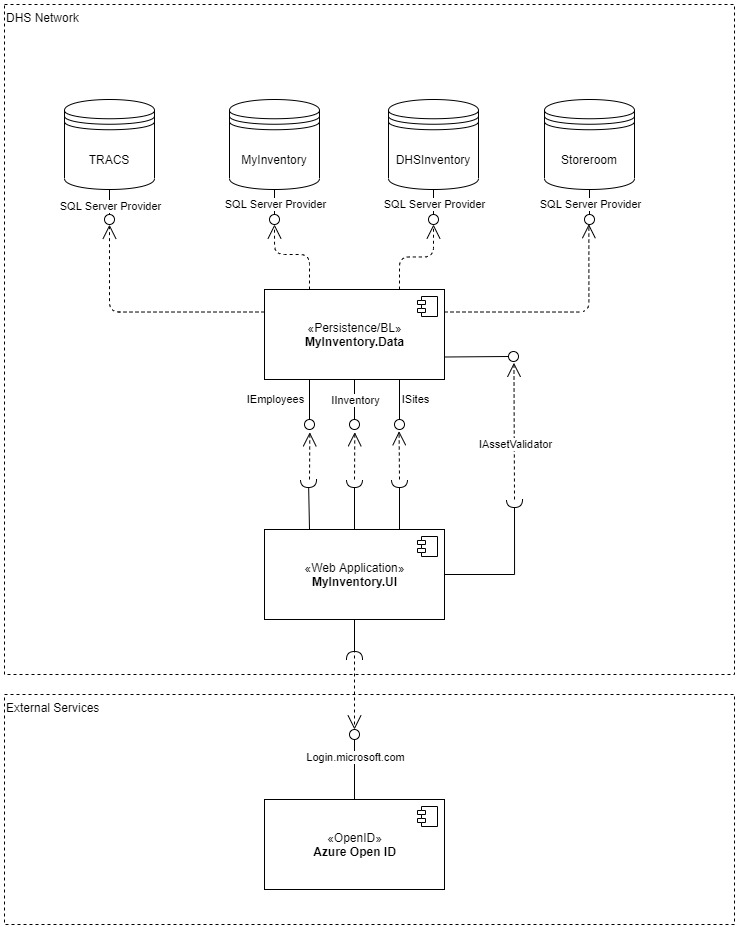


Figure : Sample UML Component Diagram

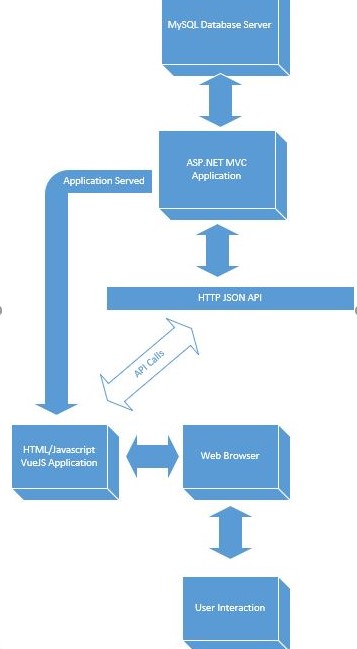


Figure Simple Block Diagram

## Classes

This section outlines the proposed classes that will be created to build the system.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Component | Description |
|  |  |  |  |
|  |  |  |  |

### UML Class Diagram

Provide a visual representation of the classes for each component

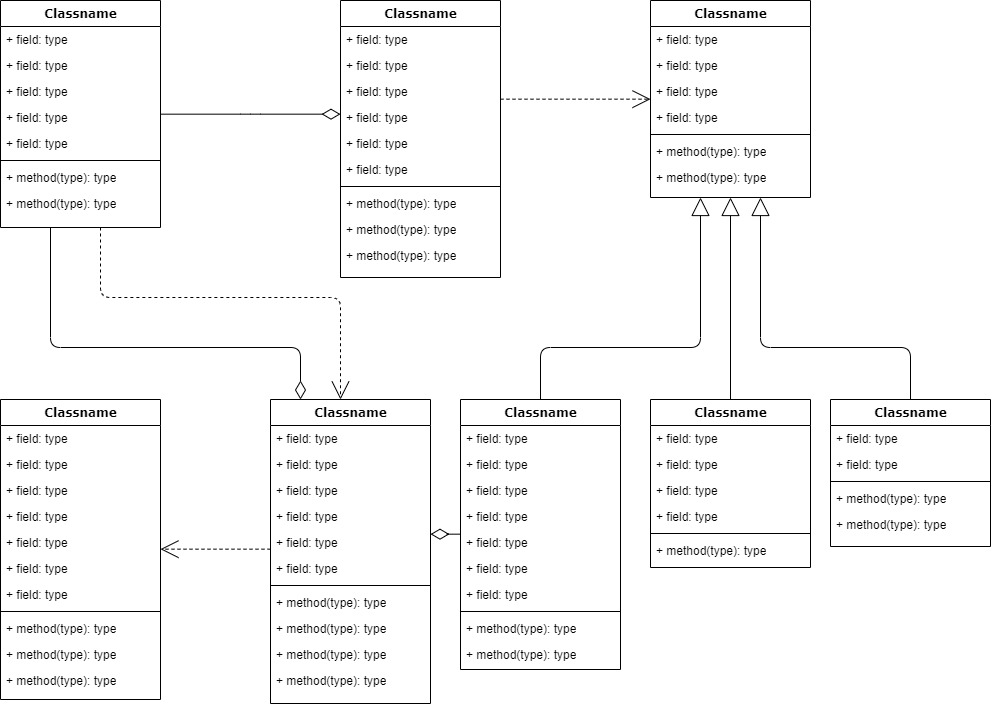


Figure Sample UML Class Diagram

## Database

Provide an overview of the database management system/environment and any special considerations regarding the database.

### Entity Relation Diagram

Provide an Entity Relation Diagram using Crows Foot or Chen notations that visually represents the database(s).

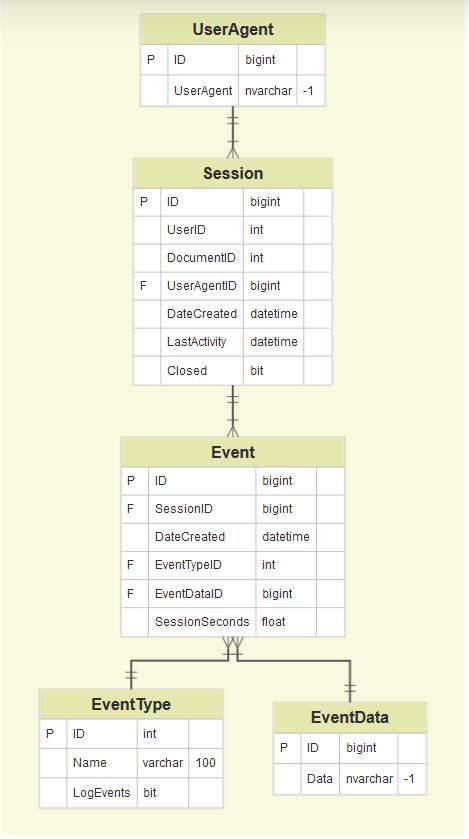


Figure Sample UML Class Diagram

# Use Cases

For each functional requirement (or closely related group of functional requirements), provide a use case and accompanying diagrams.

## User Case Title

|  |  |
| --- | --- |
| Use Case [x] |  |
| Functional Requirement(s) Mapping(s) |  |
| Actor(s) |  |
| Overview |  |
| Trigger |  |
| Pre-Conditions |  |
| Basic Flow |  |
| Alternate Flow | Do of many of these as needed |
| Post-Conditions |  |

### Visualization(s)

Include this section for each use case if applicable. Generally it is a good idea to provide at least 1 visual representation/use case.

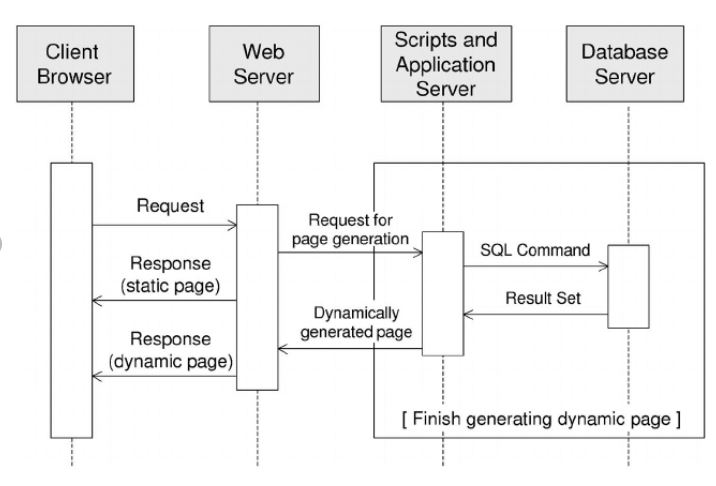


Figure Sample Sequence Diagram

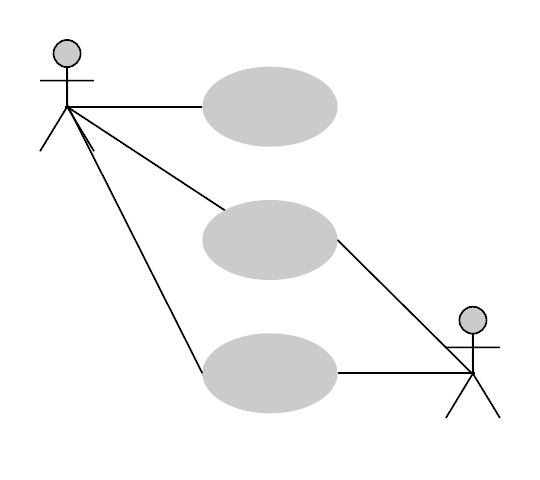


Figure Sample Use Case Diagram

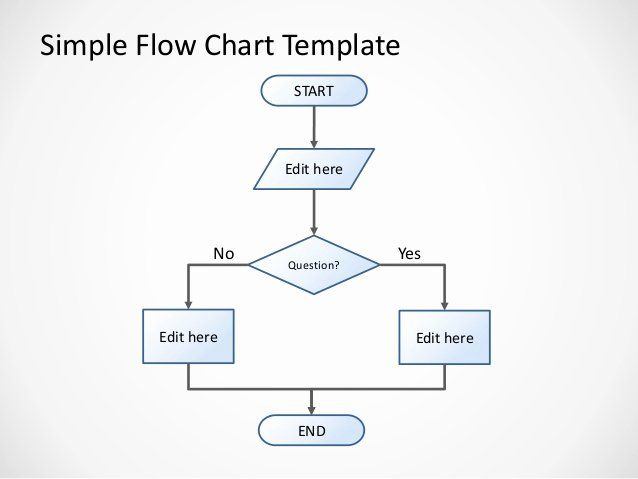


Figure Sample Flow Chart

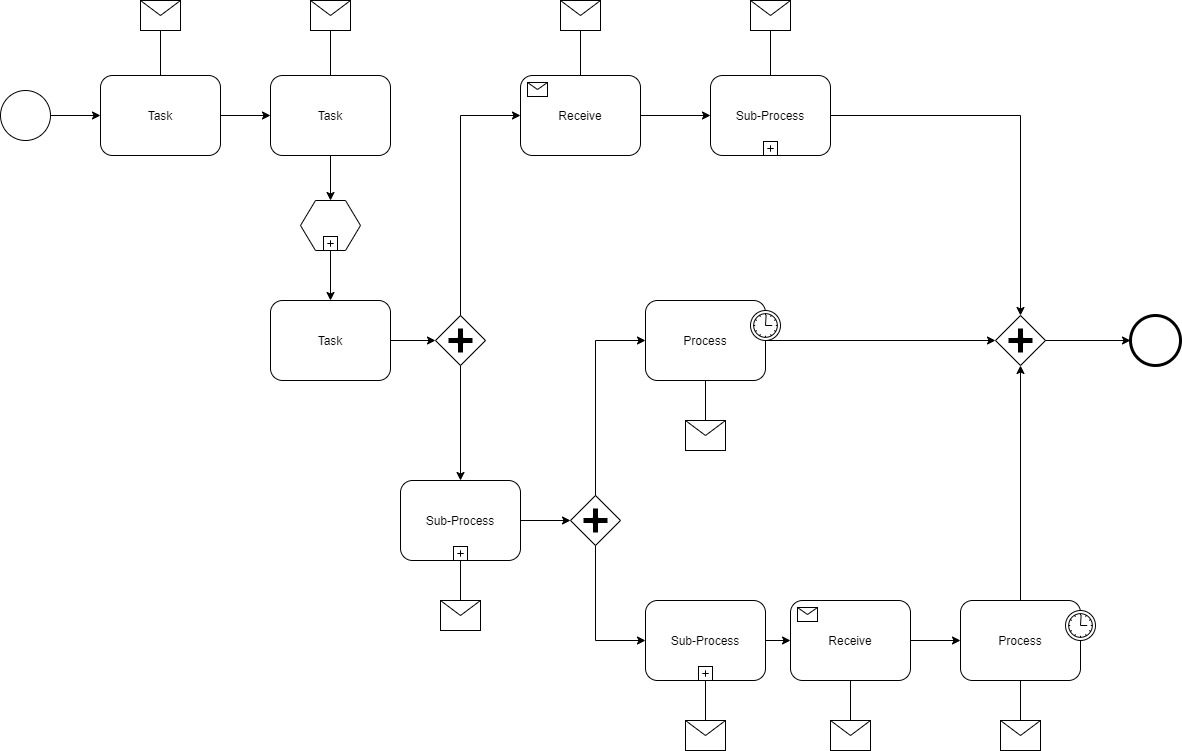


Figure Sample BPMN Diagram

# Testing

This section examines the type of testing to be done and what test cases should be performed.

## Methodology

This section examines to what extend the application needs to be tested and what methodologies will be used/steps taken to perform testing.

## Test Cases

For each use case complex enough to warrant it, develop a set of black-box test cases that can be used to verify that the application meets the system requirements. You will likely need multiple tests per-use case as you will need to explore alternate paths through the use case as well.

|  |  |
| --- | --- |
| Test Case 1 |  |
| Description |  |
| Use Case Mapping |  |
| Pre-Conditions |  |
| Inputs |  |
| Steps |  |
| Expected Result |  |

|  |  |
| --- | --- |
| Test Case 2 |  |
| Description |  |
| Use Case Mapping |  |
| Pre-Conditions |  |
| Inputs |  |
| Steps |  |
| Expected Result |  |

1. The matter of where the data should be stored is still being discussed. We will proceed with a new database, as the number of tables being created is very small; however, we may be moving these tables to an existing database before the project is deployed. [↑](#footnote-ref-2)