Table of Contents

[Requirement Analysis document 6](#_Toc448775101)

[Introduction 6](#_Toc448775102)

[Purpose of system 6](#_Toc448775103)

[Scope of system 6](#_Toc448775104)

[Objective and success criteria 6](#_Toc448775105)

[Overview 6](#_Toc448775106)

[Current System 6](#_Toc448775107)

[Proposed System 6](#_Toc448775108)

[Functional Requirements 6](#_Toc448775109)

[Nonfunctional Requirements 7](#_Toc448775110)

[System Models 7](#_Toc448775111)

[Scenarios 7](#_Toc448775112)

[Use case model 8](#_Toc448775113)

[Object model 9](#_Toc448775114)

[User interface 9](#_Toc448775115)

[System design document 10](#_Toc448775116)

[Performance Criteria 10](#_Toc448775117)

[Response Time 10](#_Toc448775118)

[Throughput 10](#_Toc448775119)

[Memory 10](#_Toc448775120)

[Dependability Criteria 10](#_Toc448775121)

[Robustness 10](#_Toc448775122)

[Reliability 10](#_Toc448775123)

[Availability 10](#_Toc448775124)

[Fault Tolerance 10](#_Toc448775125)

[Security 10](#_Toc448775126)

[Safety 11](#_Toc448775127)

[Cost criteria 11](#_Toc448775128)

[Development Cost 11](#_Toc448775129)

[Deployment Cost 11](#_Toc448775130)

[Upgrade cost 11](#_Toc448775131)

[Maintenance Cost 11](#_Toc448775132)

[Administration Cost 11](#_Toc448775133)

[Maintenance Criteria 11](#_Toc448775134)

[Extensibility 11](#_Toc448775135)

[Modifiability 11](#_Toc448775136)

[Adaptability 12](#_Toc448775137)

[Portability 12](#_Toc448775138)

[Readability 12](#_Toc448775139)

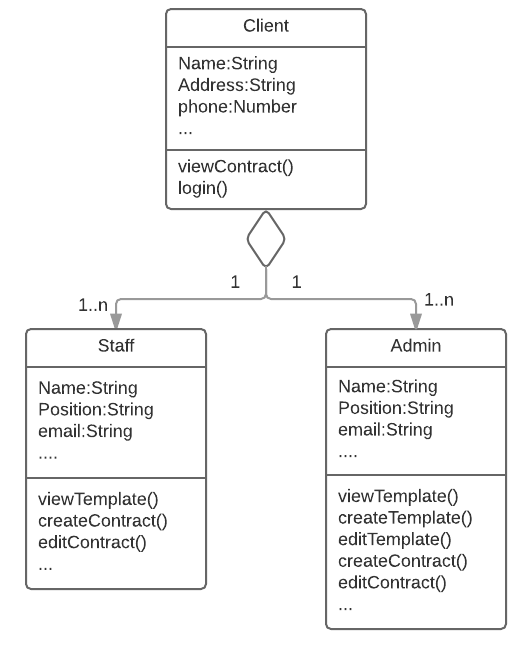
[Traceability of Requirements 12](#_Toc448775140)

[End User Criteria 12](#_Toc448775141)

[Utility 12](#_Toc448775142)

[Usability 12](#_Toc448775143)

[Class Diagram 13](#_Toc448775144)

[ 13](#_Toc448775145)

[Design Goals 13](#_Toc448775146)

[References 13](#_Toc448775147)

[Current Software Architecture 14](#_Toc448775148)

[Proposed Software Architecture 14](#_Toc448775149)

[Identify and storing persistent data 14](#_Toc448775150)

[Providing access control 14](#_Toc448775151)

[Designing the global control flow 14](#_Toc448775152)

[Identify services 14](#_Toc448775153)

[Boundary Conditions 15](#_Toc448775154)

[Object Design Document 15](#_Toc448775155)

[Object Design Trade-offs 15](#_Toc448775156)

[Interface Documentation Guidelines 15](#_Toc448775157)

[References 15](#_Toc448775158)

[Class Interface 16](#_Toc448775159)

[User manual 17](#_Toc448775160)

[Steps 17](#_Toc448775161)

[Test manual 18](#_Toc448775162)

[Steps 18](#_Toc448775163)

[Administration document 19](#_Toc448775164)

[Project Termination 19](#_Toc448775165)

[Delivery 19](#_Toc448775166)

[Postmortem 19](#_Toc448775167)

[Administration Manual 19](#_Toc448775168)

[Install 20](#_Toc448775169)

[Operate 20](#_Toc448775170)

[Bring down the system 20](#_Toc448775171)

[Error Codes 20](#_Toc448775172)

[Failure and termination conditions 21](#_Toc448775173)

[Glossary 21](#_Toc448775174)

[Conclusion 21](#_Toc448775175)

# Requirement Analysis document

## Introduction

We were tasked with creating a web application that would assist Element 74 create contracts for all their future projects. Clients will be able to view these contracts and can accept them if they are qualified.

### Purpose of system

The purpose of the system is to create and generate contracts for clients to view and accept.

### Scope of system

The web application must be completed by the end of the semester. The final product should be bug-free and should perform in an acceptable manner regarding speed and reliability.

### Objective and success criteria

* Multiple projects must be able to be added to a client, and a contact should be generated based off of the project requirements.
* A PDF version of each contract must be available for viewing and download.
* An electronic signature from the client is required for each contract.
* An email must be sent to whoever issued a contract once it is signed.
* Only administrators and those that created the contract should be able to edit it.
* Normal users can add comments to the contracts

### Overview

The contract generator that will be designed for Element 47 will be web application that will assist with the client registration, contract creation, contract template creation, and contract acceptance.

## Current System

There is not currently a system in place; it will be built from scratch.

## Proposed System

### Functional Requirements

* Multiple projects must be able to be added to a client, and a contact should be generated based off of the project requirements.
* A PDF version of each contract must be available for viewing and download.
* An electronic signature from the client is required for each contract.
* An email must be sent to whoever issued a contract once it is signed.
* Only administrators and those that created the contract should be able to edit it.
* Normal users can add comments to the contracts

### Nonfunctional Requirements

1. Usability  
   The application should be intuitive to use for each type of user (admin, client, etc.) Each action that each user can perform should be clearly established and easy to perform.
2. Performance  
   The system should have quick response times, especially for logins, contract generation, and contract viewing/downloading. The slowest portion of this application will likely involve the databases.
3. Supportability  
   The application must be flexible enough to handle multiple templates of the contract so that in future if adjustments need to be made to a contract, it can be down without any programming or logic changes.
4. Implementation  
   Implementing this project is restricted by what is compatible and what is available. The databases must be able to communicate with the webpage and the programming side must also work with the webpage.
5. Operation  
   The operability of this system is dependent on how we choose to host the page. If we are able to get a reliable hosting service, the web page should be in functioning condition %99.99 of the time.
6. Legal  
   The legal aspects will likely be handled by Element 74. However, we are tasked with keeping client information secure, which we plan to do.

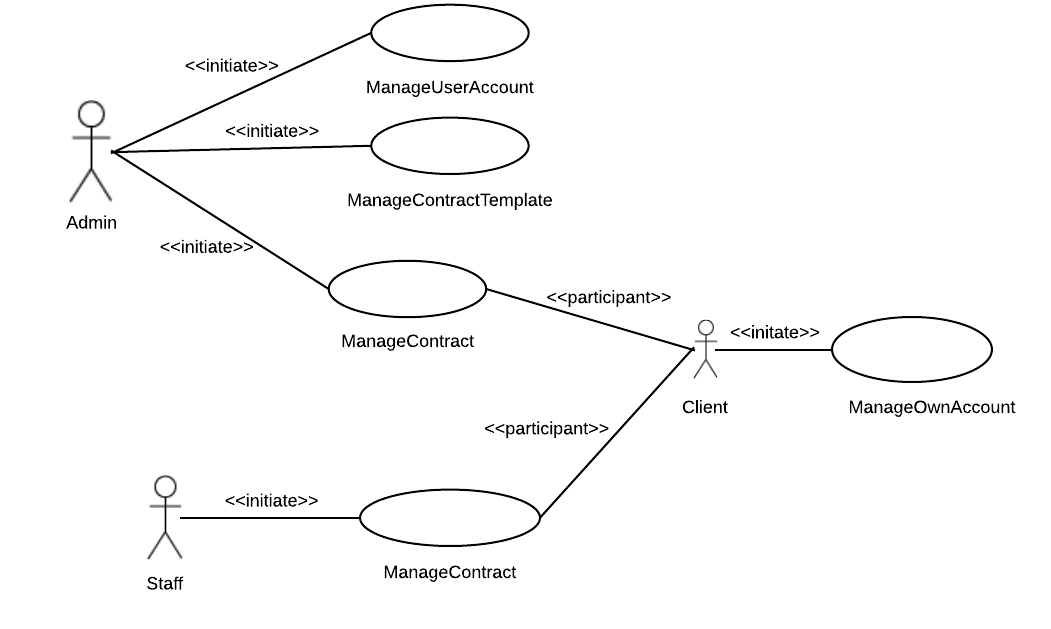
### System Models

### Scenarios

Scenario: Manage Contact  
Actors: Admin, Staff, and Client

Both the admin and staff have the ability to register a client to the system by filling in fields for their name, email, etc. After the information is provided for the client, the client can edit any information entered by the admin/staff by filling in the fields, accessed through the main menu.

### Use case model



#### Use Cases

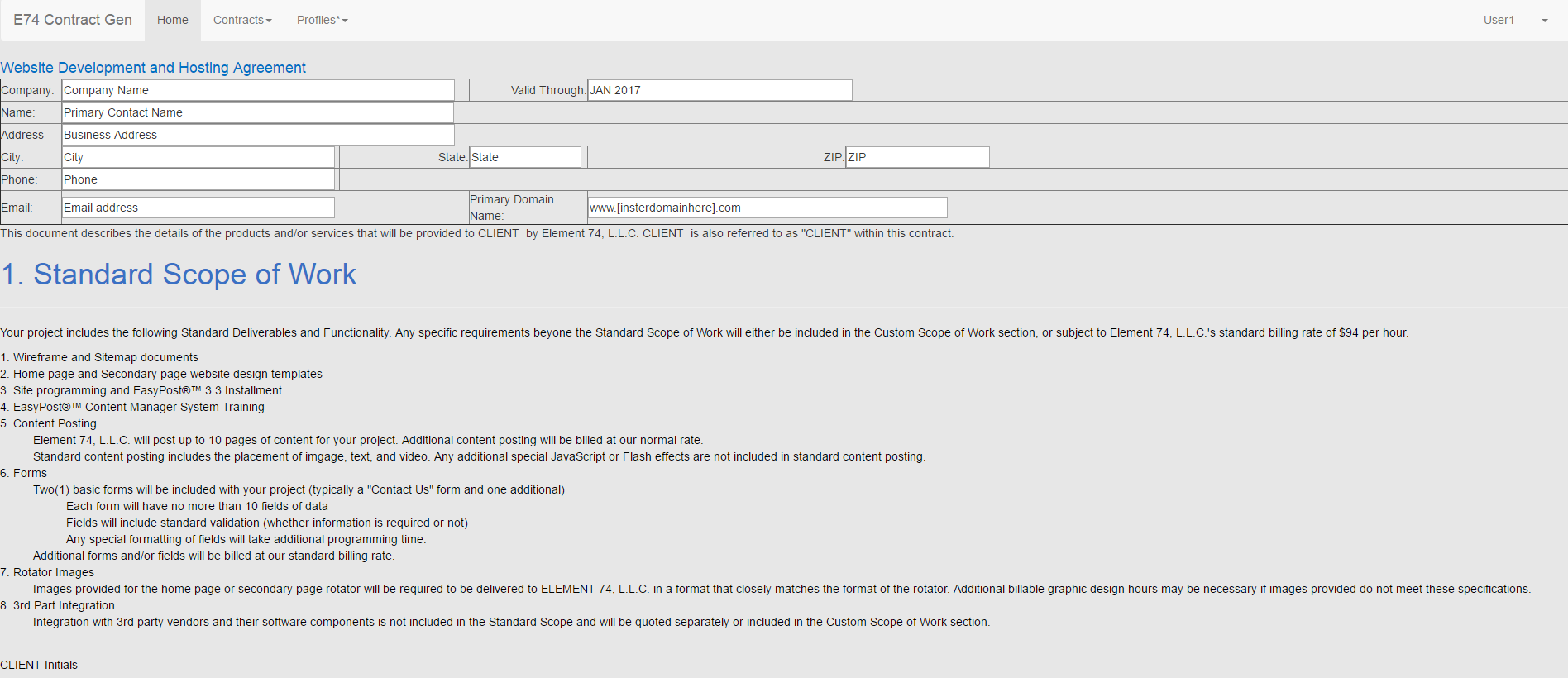
1. = Admins Only
2. = Staff
3. = Client

* Manage User Account (A)
* Manage Contract Template (A)
  + Create Template
  + Edit Template
  + View Template
* Manage Contract (A, B)
  + Create Contract
  + Edit Contact
* Manage Own Account (C)
  + Edit Profile
  + Access Contract

### Object model



### User interface



# System design document

## Performance Criteria

### Response Time

The contract initiator should be notified with 0.5 seconds of a client signing a contract.

### Throughput

The system should be able to generate 20 contracts an hour and save all information to the database.

### Memory

The database will require 120GB to store all client information

## Dependability Criteria

### Robustness

Invalid information will not be saved to database (email, address, etc.)

### Reliability

There should not be any difference between specified and observed behavior

### Availability

The system should have 99% annual availability

### Fault Tolerance

The system should be tolerant to loss of connection to server or database.

### Security

The system should be secure and not allow non-staff members to view client information.

### Safety

This system will not create safety concerns to human lives.

## Cost criteria

### Development Cost

Development is using free software.

### Deployment Cost

Deployment cost should be below $100, as the server is being provided and other services should be inexpensive.

### Upgrade cost

This system is not upgrading from a previous system.

### Maintenance Cost

Costs for bug fixes and improvements will be $20/hr.

### Administration Cost

Administration cost should be less than $200.

## Maintenance Criteria

### Extensibility

New functions and classes can be added by staff members.

### Modifiability

Functionality of this system can be modified by staff members.

### Adaptability

This system can be ported to similar application domains with minor updates.

### Portability

The website should run on most devices capable of viewing a web browser

### Readability

The code will be documented so staff can interpret it.

### Traceability of Requirements

The requirements have already been documented.

## End User Criteria

### Utility

Information will be automatically updated and saved; emails will automatically be sent to contract initiators.

### Usability

The website will modify depending on what platform the user is viewing it with.

## Class Diagram

## 

## Design Goals

The design of the system will include two interfaces. One interface will be for the desktop version of the system that will be used by operators, and the second will be a touch version available for mobile devices.

## References

Requirements analysis, and traceability information can be found in the Detailed Design Report.

## Current Software Architecture

No current system. New system will be created from scratch.

## Proposed Software Architecture

Identify and storing persistent data

Client information will be stored in the database. Information required for contract templates will also be stored in the database.

Providing access control

Access control will be managed by giving separate logins to all users. Different users have different permissions, ensuring that, for example, a client does not create a new user.

Designing the global control flow

Queries can be used to find information previously sent to the database. Subsystems will maintain a live connection with the server and database in order to stay synchronized.

Identify services

* Contract subsystem: Creates contracts from client data, lists project requirements, and generates the PDF
* Template subsystem: Provides template information
* Client subsystem: Holds client information, handles login
* User subsystem: Checks user level, edits contract information

### Boundary Conditions

Starting up and shutting down this system must be done by a staff member of Element 74. All users will be able to continue using the system until they log out or close the window, which will log them out automatically.

# Object Design Document

## Object Design Trade-offs

* Storage- Storage will be server based so there will not be any local device access to information. All information must be downloaded from the server.
* Security- Operators will only have to login once, and again after an inactivity period. This is a low security system and is intended to provide speed to the system.
* Usability- The system will be available as a desktop and mobile version for operators. The tradeoff will include limited functions on the mobile version compared to the desktop.

## Interface Documentation Guidelines

* Classes are named with nouns referring to the subsystem or activity
* Functions are named with action verb that describes an appropriate action.
* Errors will be handled or returned via an exception.

## References

Requirements analysis, and traceability information can be found in previous reports.

## Class Interface

class partcipantInfo{

public:

string name, role, UID, submit ();

int phone;

}

public class Clients

{

private Address \_clientAddressInfo = new Address();

private int \_clientID;

private int \_contractID;

private int \_projectNumber;

private char \_clientInitials;

private string \_esignature;

public Address Address{get;set;}

public int ClientID{get;set;}

public int ContractID{get;set;}

public int ProjectNumber{get;set;}

public char ClientInitials{get;set;}

public string ESignature{get;set;}

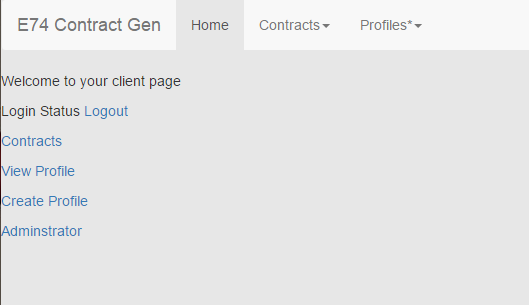
}

# User manual

## Steps

**Step 1** Users should use login information to sign into the application.

**Step 2** After sign on Users are presented with a user interface similar to the image below.



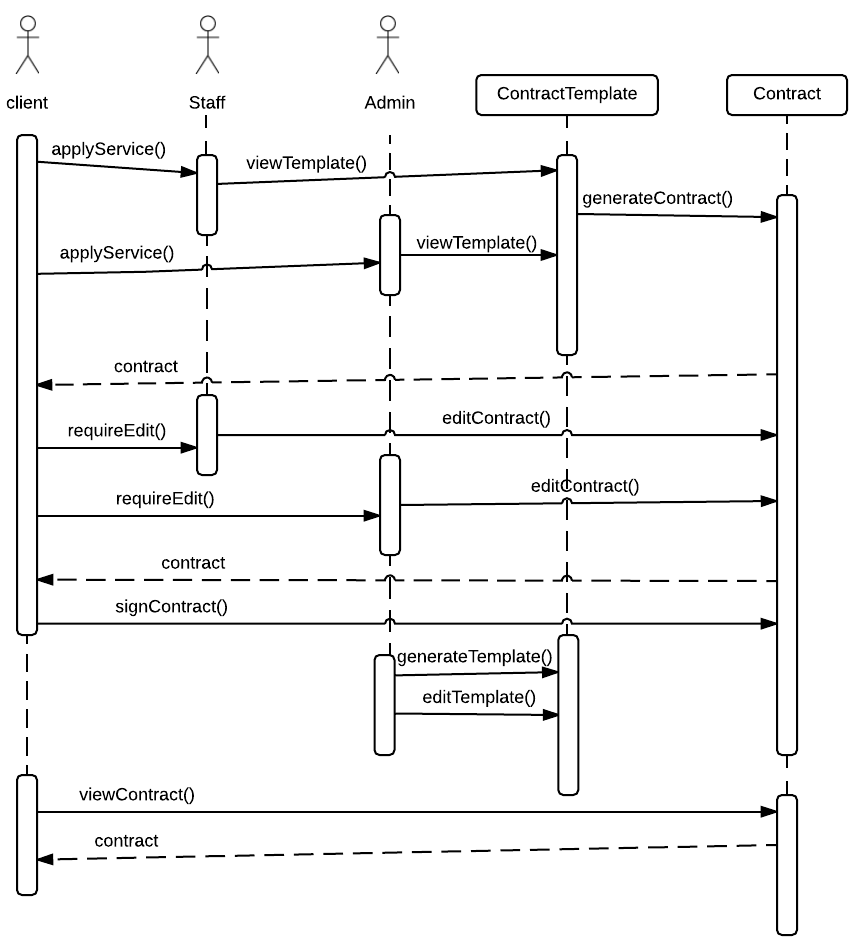
**Step 3** Operators select the appropriate tab (Home, Contracts, Profiles) from the top of the interface and enter the required information.

**Step 4** After information has been added submit to send information to server.

# Test manual

## Steps

To complete the testing process each component will be checked for program faults in source code. **Step 1** Test contract template creation by logging into the system as an admin and entering relevant information into the interface.

**Step 2** Submit template information to server to check for problems with upload operations. See diagram for flow of information. 

# Administration document

## Project Termination

Project termination will include the contract generator being delivered to the client and the system installation. A client acceptance test will be performed before product installation is complete.

### Delivery

Product delivery will consist of client acceptance test and installation.

#### Client Acceptance Letter/Form

The software system that is being evaluated with the client will be based on the acceptance criteria set in the project agreement. Functional and nonfunctional requirements are tested and evaluated, and the client will accept the product before installation.

#### Installation Test

The installation test will include a field test of the system. The tests will be conducted under real world conditions to look for faults in the system.

### Postmortem

Postmortem activities will include reviewing project history and notes, assessing unforeseen issues, and reviewing failures. The goal of the postmortem activities is for the project team to learn from problems and issues to improve the output on the next project.

## Administration Manual

The administration manual provides instructions for administration level procedures.

### Install

To install the application:

1. Extract the downloaded files to location.
2. Run the setup.exe file
3. Follow the setup wizard for installation procedures.

### Operate

To operate the application:

1. Click the desktop icon or find in application/Start menu.
2. Login as an operator
3. Gather participant information.

### Bring down the system

To manually bring down the system:

1. Press and hold the Ctrl or Command key and press the Esc key until the dialog box appears.
2. When the manual shutdown box appears, you must agree to losing all information not uploaded before program shutdown.
3. Click shutdown program.

### Error Codes

1. 101- Lost connection to server
2. 102- Low server storage
3. 103- Missing participant information

### Failure and termination conditions

The application will fail if host machine fails to allocate proper memory for the system. The program will terminate if host machine does not have proper memory to perform required tasks.

# Glossary

* Admin-User of registration system with administrative privileges.
* Client- User who accepts contracts.
* Server-Web server where participant information is stored.

# Conclusion

We are creating a web application that will create contracts for different projects. The application must securely hold client information. A PDF version of contract should be available to download. After a client digitally signs a contract, a notification is sent to the individual that initiated the contract.