Connor Petford (Student)

Connor.petford@techtorium.ac.nz

Prerequisites Document

Covers Design Consideration, Functional and Non-Functional Requirements

Software requirement specification

For Visitor Registration Application

**Contents**

[1. Introduction 2](#_Toc127950368)

[1.1 Purpose 2](#_Toc127950369)

[1.2 Intended Audience 2](#_Toc127950370)

[1.3 Intended Use 2](#_Toc127950371)

[1.4 Product Scope 2](#_Toc127950372)

[1.5 Definitions and Acronyms 2](#_Toc127950373)

[2. Design Considerations 2](#_Toc127950374)

[2.1 Operating Environment 2](#_Toc127950375)

[2.2 Design and Implementation Constraints 3](#_Toc127950376)

[2.3 Assumptions and Dependencies 3](#_Toc127950377)

[3. Functional Requirements 3](#_Toc127950378)

[3.1 User Interface 3](#_Toc127950379)

[3.2 Software Interface 3](#_Toc127950380)

[4. Non-Functional Requirements 4](#_Toc127950381)

[4.1 Performance Requirements 4](#_Toc127950382)

[4.2 Security Requirements 4](#_Toc127950383)

**Software Requirement Specification**

# Introduction

## 1.1 Purpose

The purpose of this document is to provide an overview on our visitor registration desktop application. This application is designed to create a more efficient process for visitors signing in to scheduled meetings. We will implement this application with the use of the programming language C# in Microsoft Visual Studio.

## 1.2 Intended Audience

This project is intended for visitors of a company to easily sign in and display scheduled meetings, and for the company to track in progress meetings. This can range from students looking for a career pathway with the company, or regular attendees who have scheduled meetings with a staff member. It has been implemented under the specification that visitors’ data will not be stored on any systems once the meetings are over, and their personal data is wiped following their exit, for security purposes.

## 1.3 Intended Use

We’ve implemented an application that allows for a more efficient way for visitors to sign in to meetings they’ve scheduled with the company. In using our application, the company is able to track in progress meetings and clear them with ease, increasing productivity and efficiency within the workspace.

## 1.4 Product Scope

By providing our application to visitors, we allow the companies visitors to sign in and attend their meetings without having to call in or write down their personal details on a clipboard to confirm their attendance and enables the process of attending and exiting their meeting with ease. We are able to implement this as our system was easily deployed with the help of Microsoft Visual Studio.

## 1.5 Definitions and Acronyms

|  |  |
| --- | --- |
| SRS | Software Requirement Specification |
| VRA | Visitor Registration Application |
| VS | Visual Studio |

# Design Considerations

## 2.1 Operating Environment

Our software is intended to be used at the reception of (given) company, and is supported on all versions of Windows, including 11, Android, IOS and Mac systems. This allows our software to be compatible with the devices that may be used at reception, regardless of whether they change operating systems or upgrade existing systems.

## 2.2 Design and Implementation Constraints

Our VRA stores the following information during our clients visit:

* **Customer Details:**  
  This includes the customer’s name, email address and phone number. This information may be used for keeping the records of the customer for any emergency or for any other kind of information.
* **Meeting description:**  
  It includes customer details, specific times and dates of meetings, who they are meeting with and the aim of their meeting.

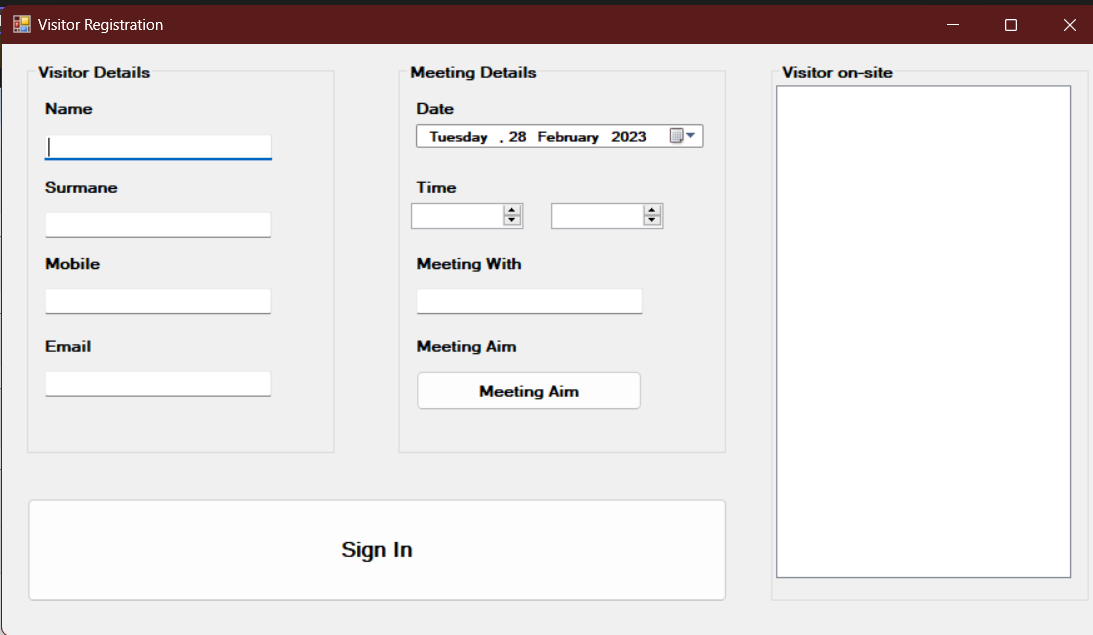
## 2.3 Assumptions and Dependencies

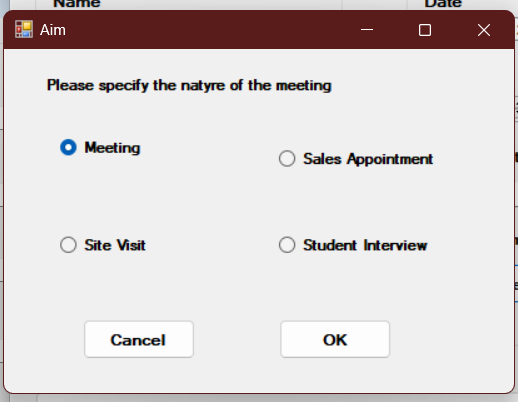
This application is to be used under the assumption that the company already handles and operates their meetings, as our software is intended only to increase the efficiency of the sign in process. It also depends on the company having access to laptops/desktops that are able to run Windows 7 or later.

# Functional Requirements

## User Interface

We’ve designed our software to meet the needs of our customers, and built a user-friendly, easy to navigate system where all the form fields are clearly set out in order, with dropdowns consisting of the company’s staff that take meetings, and an easy to access popup form where customers can state the aim of their meeting efficiently.





(Image of our application from the user’s perspective)

## 3.2 Software Interface

Our system was built using the programming language C#, in Microsoft Visual Studio, using Windows Forms with long term .NET.6 Framework support. This is to ensure the longevity and reliability of our application, and the security of our users personal data.

# Non-Functional Requirements

## 4.1 Performance Requirements

*Here we will specify the minimum system requirements to run our application (considering the host device will also be in operation by receptionist)*

CPU: Intel Core i3 3210 | AMD A8 7600 APU or equivalent

RAM: 4 GB RAM

GPU: Intel HD Graphics 4000 or AMD Radeon R5 series | NVIDIA GeForce 400 Series or AMD Radeon HD 7000 series

OS: 64-bit Windows 7 or later

Screen Resolution: 1024 x 768 or better

*Here we will specify the recommended system requirements to run our application.*

CPU: Intel Core i5 4690 | AMD A10 7800 or equivalent

RAM: 8 GB RAM

GPU: NVIDIA GeForce 700 Series | AMD Radeon Rx 200 Series

OS: 64-bit Windows 10

Screen Resolution: 1024 x 768 or better

## 4.2 Security Requirements

To ensure the privacy of customers personal data is not invaded, we have implemented a function to clear the fields containing their personal data, and displaying only the essential meeting details to the receptionist so that they are able to be efficiently taken into their scheduled meeting. The receptionist is then able to remove the meeting details once the meeting has ended, Our software also does not interfere with antivirus systems, so there is no threat of having to disable the devices antivirus protection programs