**System Requirements and Design**



Attendance System

Table of Contents

Table of Contents 3

Nomenclature 4

Preface 5

1 System Overview 6

1.1 Introduction 6

1.2 System Description 6

2 Technical Requirements 9

2.1 Functional Requirements 9

2.1.1 System 9

2.1.2 User 9

2.1.3 Attendee 9

2.1.4 Host 9

2.1.5 Appointment 10

2.1.6 Reading Terminal 10

2.2 Non-Functional Requirements 10

2.3 Graphical User Interface Specification 10

2.4 Data Communication Between Modules 12

3 System Architecture 14

3.1 Three-Tier Multi-layer Structure 14

3.1.1 Presentation Layer 14

3.1.2 Business Layer 14

3.1.3 Data Access Layer 14

4 Database 15

4.1 Model and Description 15

5 UML 16

5.1 Use Case Diagrams 16

5.2 Sequence Diagrams 16

5.3 Class Diagrams 16

# Nomenclature

RFID – Radio Frequency Identification

GUI – Graphical User Interface

PC – Personal Computer

SRS – Software Requirements Specifications

SDD – Software Design Documents

GPIO – General-purpose input/output

RS232 – Recommended Standard number 232

# Preface

This document is a combination of two documents which are normally separated; Software Requirements Specifications and Software Design Documents. This merger has been decided due to the relatively small size of the project, and will provide both the development team and stakeholders with a better overview.

The reader will get an overview of how the system architecture, database model, software architecture and user interface for the system will be arranged and implemented.

|  |  |  |
| --- | --- | --- |
| **Document version** | **Dated** | **Whats new?** |
| 1.0 | 17.01.2017 | Document created |
| 1.1 | 24.01.2017 | Added flowcharts and figures, minor structural changes |
|  |  |  |

# Introduction

Checkpoint will be a computer based attendance system capable of registering and verifying attendance to a wide range of event types. This could include students attending obligatory course events or workplaces holding compulsory meetings for employees. Often, such systems are implemented with attendances being manually recorded on paper. This approach can be both error prone and time consuming especially when collating data and calculating individual levels of attendance for larger events.

The application is intended to reduce the time needed to record attendance and provide a more convenient and reliable way to create a report from the data collected.

# System Overview

This chapter will give a brief description of the Checkpoint system.

## Introduction

## System Description

An overview of the general system structure is illustrated in *Figure 1: General system structure.* With the exception of the attendee physically attending an event, all validation and communication between the different sections of the system happens through the web application and the online database/server. A more detailed overview of the system flow can be seen in *Figure 2: Detailed System Flow*

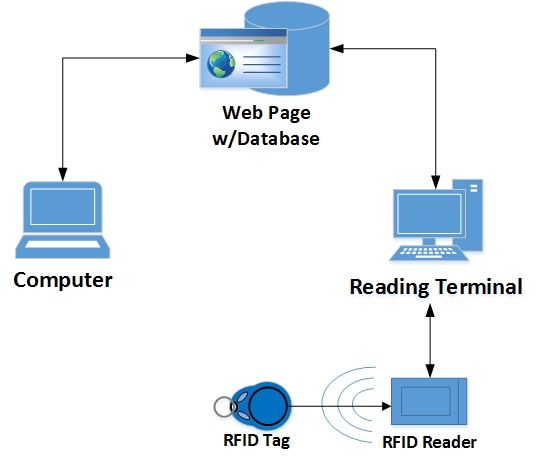


Figure ‑: General system structure



Figure ‑: Detailed System Flow

# Technical Requirements

The system requirements can be divided into two main groups: functional and non-functional requirements.

Checkpoint will have two main types of end user. A 'Host' that is a registered customer of the system with permissions to administer appointments and register attendees and a 'User' which is the status given to a subscriber who has been registered in the system and has received an RFID tag.  These are separate roles and they do not overlap, that is a Host is not a user with elevated privileges and a User cannot extend User functions to include those of a Host.

## Functional Requirements

These requirements are essential to the functionality of the software and will transform into use cases, which are described in *7.1 Use Case Diagrams.*

### System

The system requirements are those that are not restricted to a particular user but are basic general conditions that must be satisfied.

* The system must be a web-based application.
* The system must support being accessed by multiple simultaneous users without disruption.
* The system must be able to run on Google Chrome and Safari web browsers.
* The system must be able to create/read/update/delete a user or host.

### User

The following requirements are those that are specific to the user client that is defined as a person who has registered to be a potential attendee and has been allocated an RFID tag and user account.

* A user must be identified by their unique RFID tag number.
* All data entry must be validated to avoid duplicate identification fields.
* In the event that data entry validation is unsuccessful, the user must be provided with the necessary feedback via the webpage.
* A user must log in to the system with a username and password.
* A user must be able to log out of the system.
* A user must be able to change their login details.
* A user is required to submit a username, first name, last name and email.
* A user must have the ability to edit their details whenever necessary.

### Attendee

The following requirements apply to the attendee clients.

* A user becomes an attendee when he or she is granted permission to attend an appointment by a host.
* An attendee must be able to log their attendance at an appointment with his or her RFID tag.

### Host

* A host must be able to create/read/update/delete a new appointment.
* A host must be able to add users to and remove attendees from appointments.
* A host must log in to the system through the website with a username and a password.
* A host is required to submit a username, first name, last name and email.
* A host must be able to log out of the system.
* A host must be able to have a picture on his profile.
* A host must not have the ability to edit the attendance logs.
* A host must be able to request/ order an RFID reader.
* A host must be able to generate a customizable attendance report.

### Appointment

* An appointment must be created with a name, location, date, duration and the name and email of the host.
* It must be possible to define an appointment to be a single event or a series of events.

### Reading Terminal

* The reading terminal must be composed of an RFID reader and .NET compatible PC.
* The reading terminal must be able to store registrations locally to a text file as backup.
* The reading terminal must be able to connect to the internet and communicate with the server.
* The reading terminal must display the validation status on a screen to the attendee.

## Non-Functional Requirements

These requirements are related to performance and convenience, and are equally subject to testing.

* The reading terminal should be responsive enough to process and attendance within 4 seconds.
* The system should have an uptime of 22 hours per day.
* The GUI should not require the user to have any prior technical experience in order to navigate the system.
* The database storage should be easily scalable to meet future demands of a dynamic user base.

## Graphical User Interface Specification

The GUI will be a webpage accessed through a browser and is the primary point for users and hosts to interact with the CheckPoint System. All available functions should be self-explanatory and made simple to navigate. Included is a set of first drafts for the GUI, found in *Figure 3: Draft of home page*, *Figure 4: Draft of login page* and *Figure 5: Draft of user panel.*

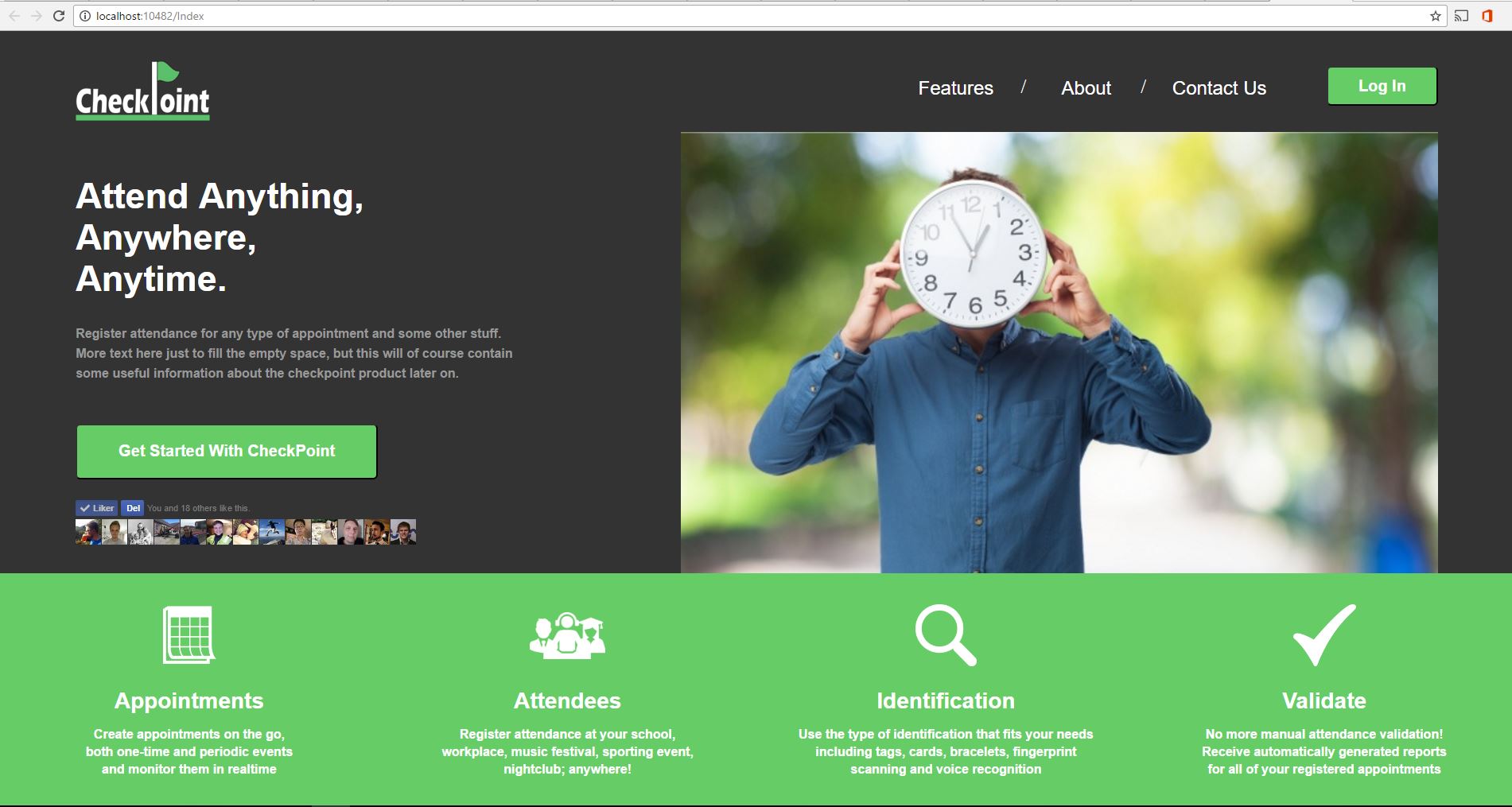


Figure 3‑1: Draft of home page

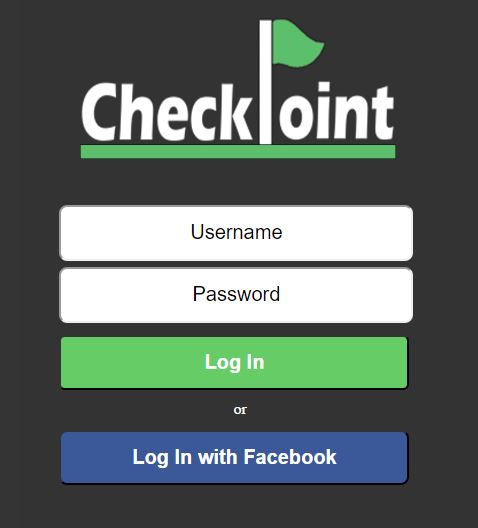


Figure 3‑2: Draft of login page



Figure 3‑3: Draft of user panel

## Data Communication Between Modules

The data communication flow between the different modules in the system is illustrated in *Figure 6: Communication between modules.*

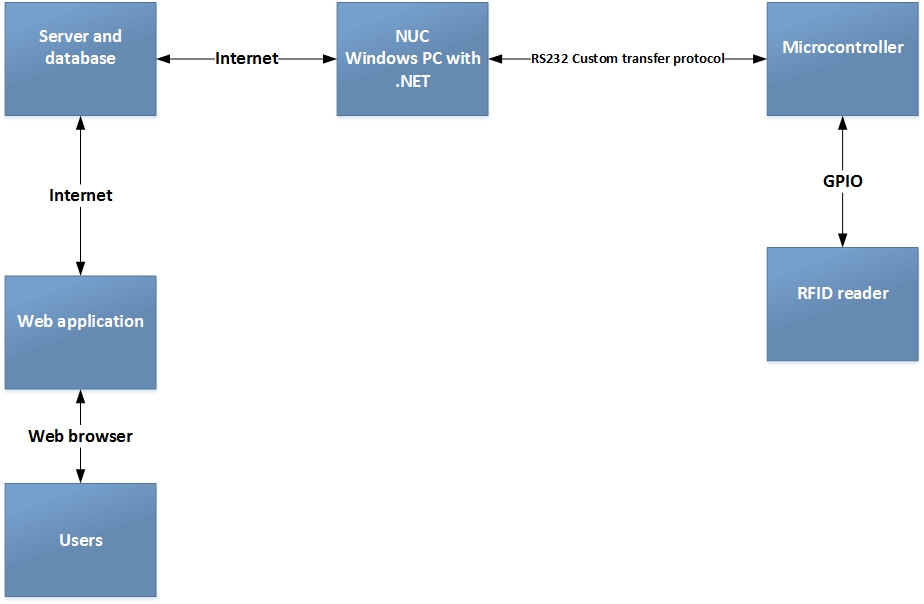


Figure ‑: Communication between modules

# System Architecture

The following chapter will give a brief explanation of the general system architecture of the CheckPoint system.

## Three-Tier Multi-layer Structure

The CheckPoint system will adhere to a 3-tier structure. This involves different parts of the application running as separate entities either physically or logically in different locations. In the case of the CheckPoint system, this will mean having a separate database server, web application server and browser. The primary aim of this structure is to provide flexibility and easier maintenance since each layer may be modified without it being necessary to make changes throughout entire application. Moreover, a layered structure provides an easier pathway to scalability and performance since it is possible to dedicate extra resources only where they are needed instead of globally. If more servers were required to handle increasing web requests this could be achieved without adding extra servers to the database or web-server application layers. Figure 4-1 below shows the basic layout of a 3-tier system.



Figure ‑ 3-Tier System Diagram

### Presentation Layer

The top tier is often referred to as the presentation layer. This layer is the point of interaction between the system and the client and includes the UI of the application. The main role of the presentation layer is to provide a convenient user-friendly way to use the application and access the functionality that the system affords. The presentation layer of the Checkpoint system will primarily consist of an ASP.NET Webform application. This will be delivered to a browser such as Google Chrome or Mozilla Firefox running on a client’s individual desktop PC or laptop. Additionally, the Checkpoint system will includes an RFID reading terminal comprising a NUC, Arduino-UNO and an RFID scanner. This part of the system will be a point of interaction at the location of all appointments. The terminals will facilitate scanning the attendees RFID tags and will provide feedback on status of the scanning procedure.

### Business Layer

The business layer is responsible for the main logic of the application. Data handling, calculations, processing requests and co-ordination of the application are all tasks performed by the business layer. It is effectively the brain of the system translating the user requests into actions and defining what operations can and cannot be performed on the data stored in the system. The CheckPoint business layer will be a C# .NET program capable of interacting with the ASP.NET Webform presentation layer and the database in the data access layer.

### Data Access Layer

The Data Access Layer describes the part of the application that communicates with the persistent storage location, usually a database server. It contains the services capable of collecting and transferring data between the storage facility and delivering them to the business layer. All interaction with the database should take place in this layer such that the other layers are unaware of the type of database or storage system being used. SQL Management Server will be the data access layer of the CheckPoint system. Figure 4-2 beneath shows a basic graphical overview of the CheckPoint system architecture.



Figure ‑ CheckPoint Architecture Overview

# Database

## Model and Description

# UML

## Use Case Diagrams

## Sequence Diagrams

## Class Diagrams