

System Requirements and Design



Attendance System

TABLE OF CONTENTS

Table of Contents	3
1 Nomenclature.....	4
2 Preface	5
3 System Overview	6
3.1 Introduction.....	6
3.2 System Description.....	6
4 Technical Requirements	9
4.1 Functional Requirements	9
4.1.1 System	9
4.1.2 User	9
4.1.3 Attendee.....	9
4.1.4 Host.....	9
4.1.5 Appointment.....	9
4.1.6 Reading Terminal.....	10
4.2 Non-Functional Requirements.....	10
4.3 Graphical User Interface Specification.....	10
4.4 Data Communication Between Modules	12
5 System Architecture.....	13
5.1 Three-Tier Multi-layer Structure.....	13
5.1.1 Presentation Layer.....	13
5.1.2 Business Layer.....	13
5.1.3 Data Access Layer	13
6 Database	14
6.1 Model and Description.....	14
7 UML	15
7.1 Use Case Diagrams	15
7.2 Sequence Diagrams	15
7.3 Class Diagrams.....	15

1 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

2 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

3 SYSTEM OVERVIEW

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

3.1 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.

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.

3.2 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: System flow*

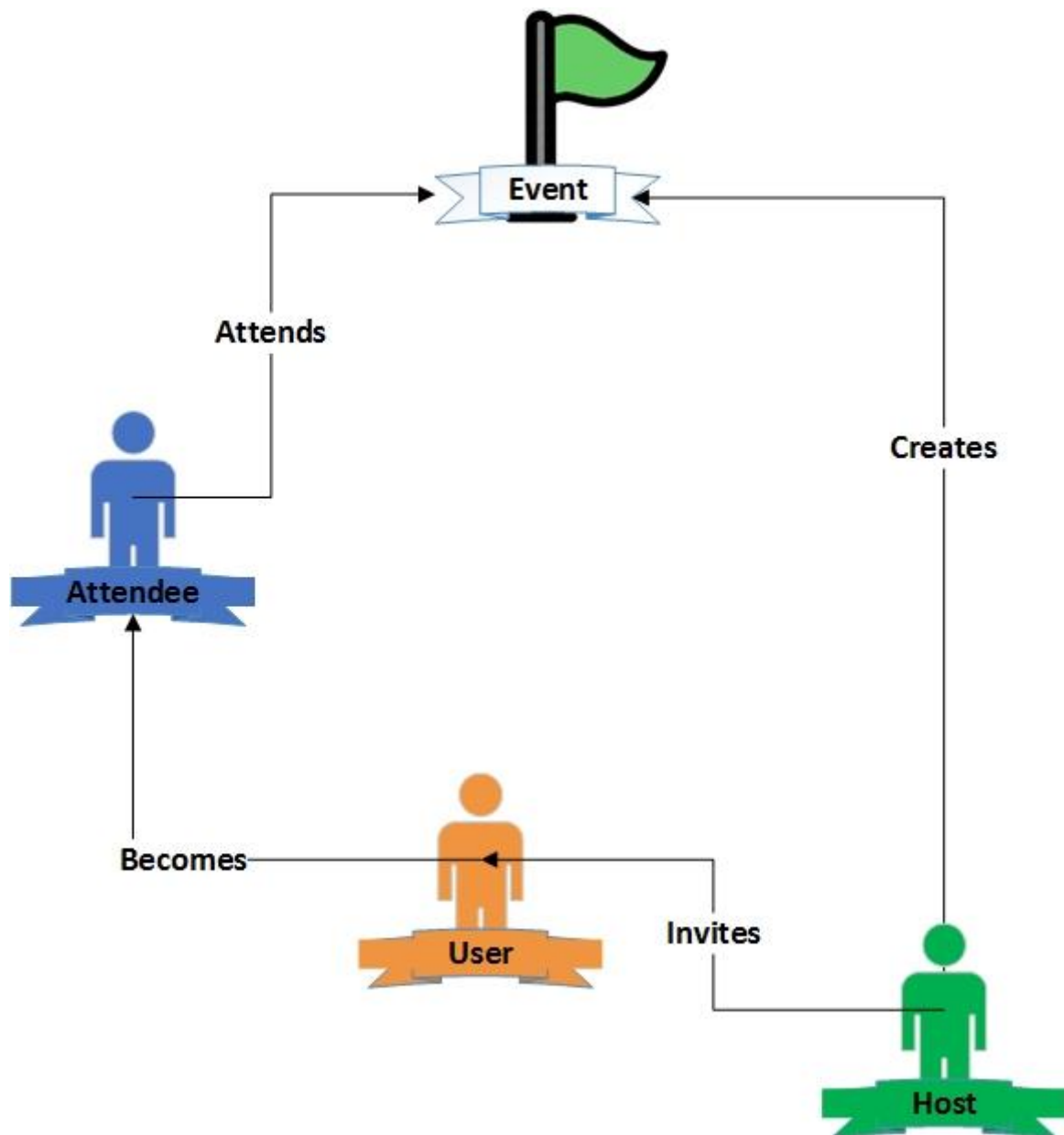


Figure 1: General system structure

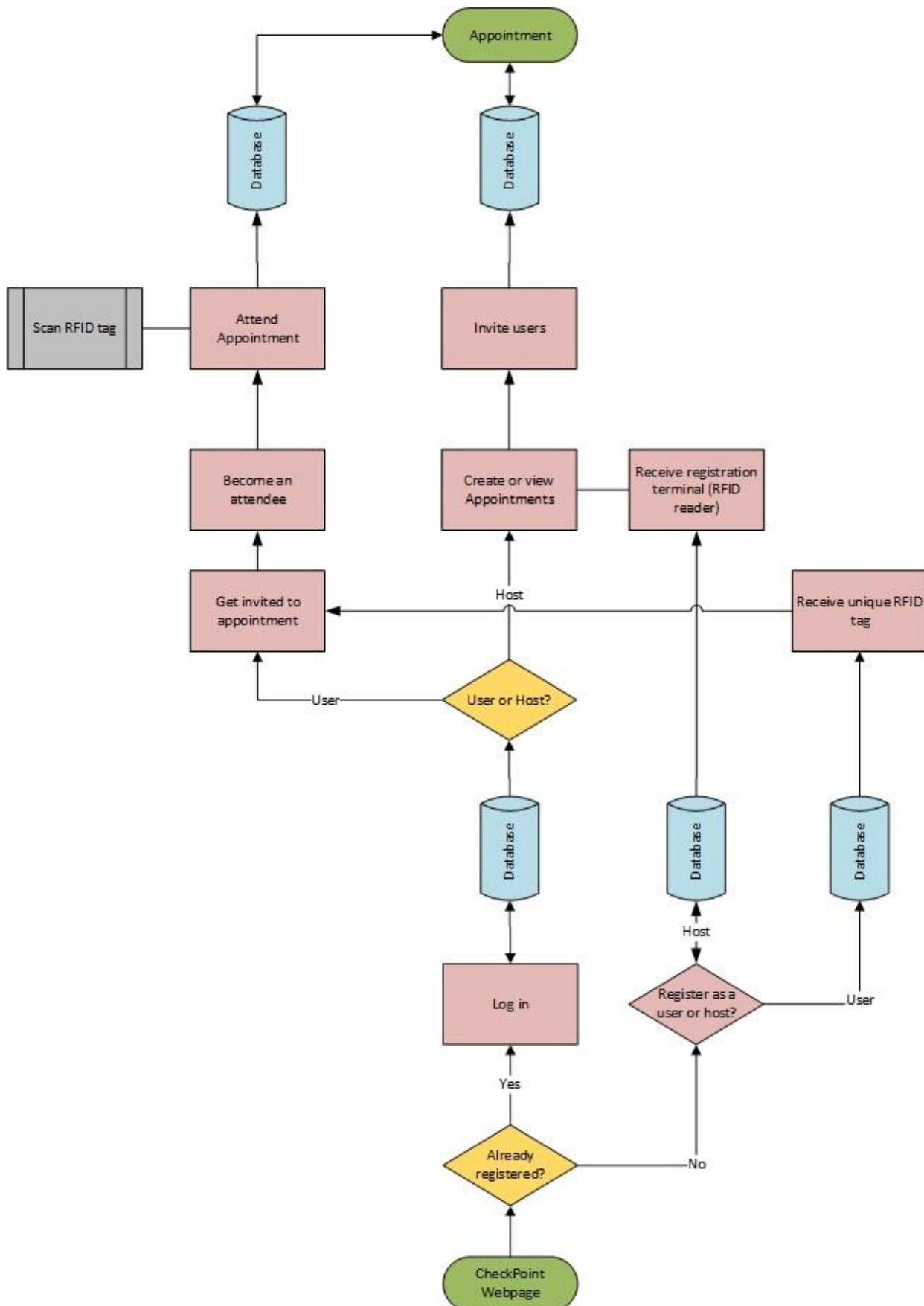


Figure 2: System flow

4 TECHNICAL REQUIREMENTS

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

4.1 Functional Requirements

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

4.1.1 System

- 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.

4.1.2 User

- Each user of the system must be uniquely identified by his or her 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 about this through the webpage.
- A user must log in to the system with an email and password.
- A user must be able to change their login details.
- A user is required to submit a first name, last name, email, address and phone number.
- A user must have the ability to edit their details whenever necessary.

4.1.3 Attendee

- 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.

4.1.4 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 an email address and a password.
- 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.

4.1.5 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.

4.1.6 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.

4.2 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 2 seconds.
- The system should have an uptime of 24 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.

4.3 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*.

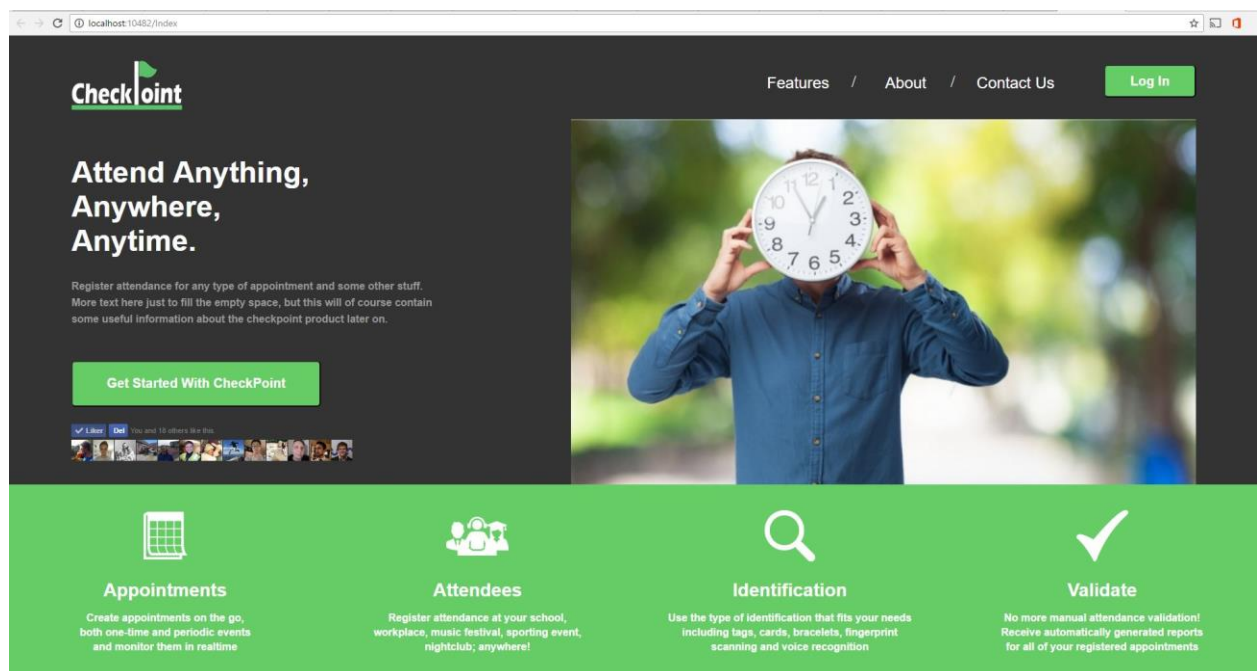
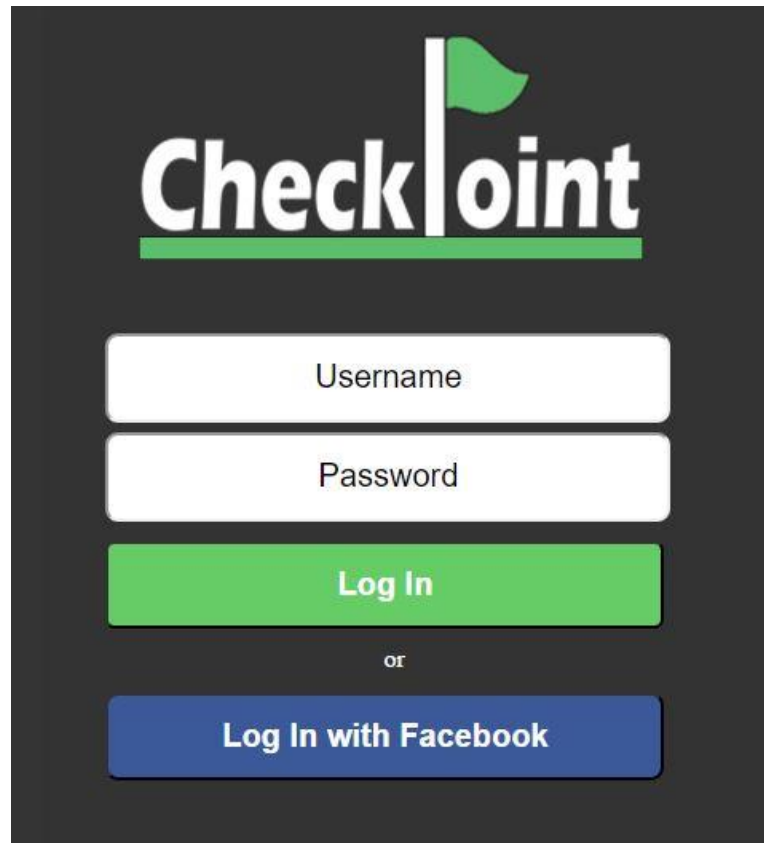


Figure 3: Draft of home page



The login page features a dark gray background. At the top center is the 'Checkpoint' logo, where the 'i' in 'Point' is replaced by a green flag on a white pole. Below the logo are two white input fields: 'Username' and 'Password'. Under these is a green 'Log In' button. Below the button is the text 'OR' in white. At the bottom is a blue button with white text that says 'Log In with Facebook'.

Figure 4: Draft of login page



The user panel is displayed in a web browser window at localhost:10482/UserPanel. It has a green sidebar on the left and a main content area on the right.

Sidebar (Left):

- A green box containing a white Batman silhouette.
- Text: "Hi, im..."
- Bruce Wayne**
- Text: "And I live at..."
- Text: "1007 Mountain Drive, Gotham"
- At the bottom: "Alter Personality?"

Main Content Area (Right):

Appointment	Who	Where	When	What else?
Lab-assignment	Prof. Leonard	C-325	28/01 - 11.15	Remember notes
Offspring Concert	RockClub 49	1036th Blv. Ave.	02/02 - 22.15	Don't get too drunk
Pokemon GO meet-up	East Valley Pokestars	St. Mayfield Park	04/02 - 12.00	Extra powerbank
Fighting Crime	The Joker	Arkham Hospital	17/03 - 08.30	Get Mark Hamill poster signed

Below the table is a green button with white text: "Look at already attended appointments".

Figure 5: Draft of user panel

4.4 Data Communication Between Modules

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

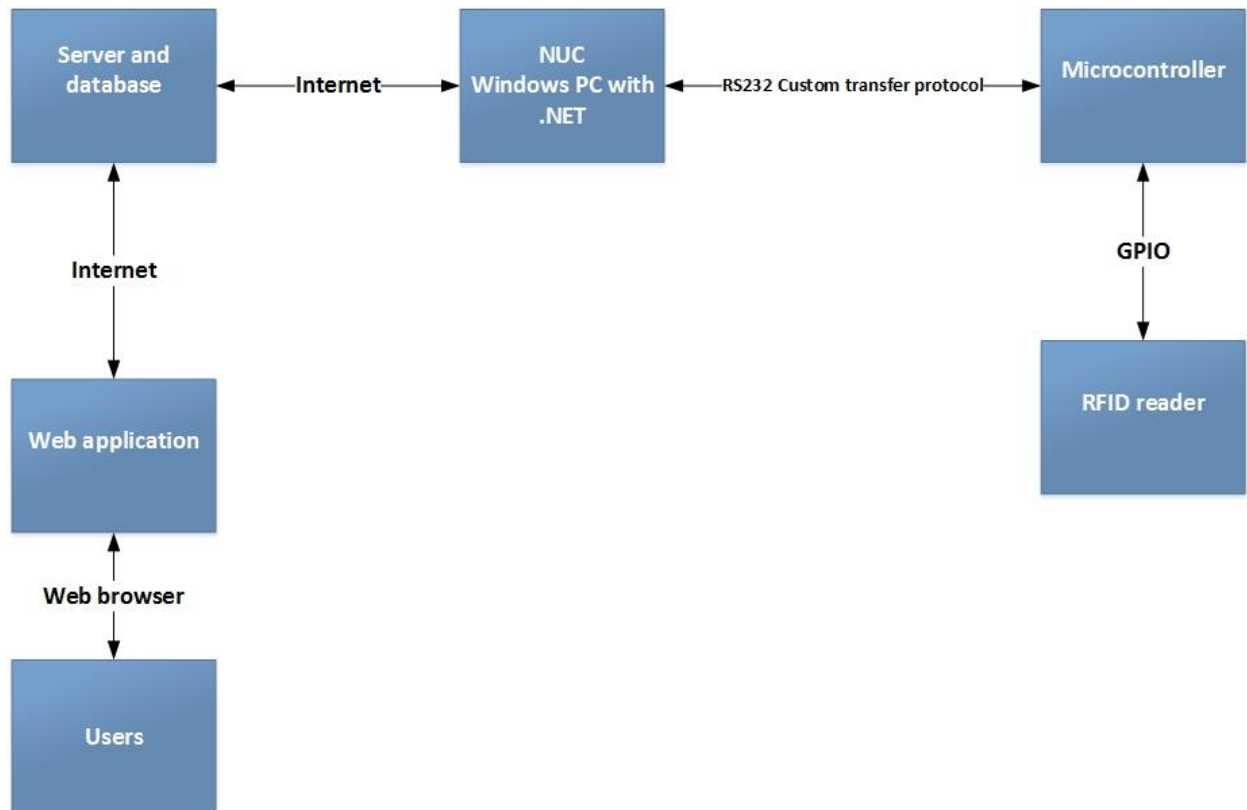


Figure 6: Communication between modules

5 SYSTEM ARCHITECTURE

5.1 Three-Tier Multi-layer Structure

5.1.1 Presentation Layer

5.1.2 Business Layer

5.1.3 Data Access Layer

6 DATABASE

6.1 Model and Description

7 UML

7.1 Use Case Diagrams

7.2 Sequence Diagrams

7.3 Class Diagrams