1 Architecture requirements

1.1 Architectural scope

The project will consist of five main components of which one is still a proof of concept (POC).

- Web Administration
- Mobile Application
- Web API
- Basic Reporting
- Embedded Hardware and prototyping (POC)

1.2 Quality requirements

The quality requirements are the requirements around the quality attributes of the systems and the services it provides. This includes requirements like performance, scalability, security, auditabilty, usability, and testability requirements.

1.3 Integration and access channel requirements

There are no external systems that must be integrated with. However, it needs to integrate with the various hardware devices (such as RFID transmitters and receivers).
The system will be access in two ways; the mobile app, which is used by the parents that are monitoring their children and through a web portal, which is how administrators will access the system for maintenance.

1.4 Architectural constraints

The only constraint for this project is the tools used to build the database for the system, those are the tools the client prefers and specified in the requirements specification.

2 Architectural patterns or styles

The main programming language to be used is C#, since we are required to use the .NET Framework.
The client browsers and app will be using HTTP messages to communicate with the web API.
The client requested that we make use of the Entity Framework, which is an ORM (object-relational mapper), as it will greatly improve maintainability.

3 Architectural tactics or strategies

As for Architectural tactics and strategies, we will are proposing to use a multi threaded approach in order to achieve scalability. Caching will be done on the maps that need to be downloaded to the mobile application, this will allow users to keep the maps of areas they feel they will frequently visit.

4 Use of reference architectures and frameworks

We will be using an object relational mapper in the form of the Entity Framework. This will be used to map to the SQL database making accessing database values easier through objects in the code. Reasons for choosing the Entity Framework is that in the type of database access we are dealing with, this object relational mapper provides reuse-ability so that if changes need to be made these changes are easier to implement in the future.

5 Access and integration channels

End product users will access the system through a Mobile application and the administrators of the system will access the system through a Web API with more privileges and higher level of access of the system compared to the average user. Due to the nature of the project and technologies that will be used, integration is fairly due to the fact that the programming languages for most components are all compatible.

6 Technologies

The technologies that will be used for this project are the following:

- Microsoft Visio Studio
- ASP .Net MVC 5
- HTML 5
- JavaScript
- JQuery
- ASP .Net WebApi
- C#
- Microsoft SQL Server 2012
- Xamarin

The above mentioned technologies are the technologies that are to be used for this project, but may still be subject to change at a later stage of the software development due to the unforeseen situations.

Constraints

1 Functional requirements and application design

1.1 Introduction

This section discusses the functional requirements for the KinderFinder System, this introduces the core domain concepts and relationships between these concepts.

The requirement is to create a system, that will through the use of RFID technology in wristbands and matching readers / receivers, be able to track children throughout a restaurant or designated area. This information needs to be fed to an API on a server that will push information to the relevant devices (mobile) to inform parents of the children's whereabouts within the designated area.

1.2 Required functionality

1.2.1 Web Administration

The Web administration must:

- Be used to allow installers of the system to set-up restaurants and their layouts (using maps).
- Enable restaurants to link wristbands to a Patron's mobile application, or assigned electronic tracking device (device not in scope).
- Enable restaurants to clear and unlink wristbands from devices and patrons' mobile apps.
- Allow access to basic reporting will also be done through the Web Administration.

1.2.2 Mobile Application

The mobile implementation must:

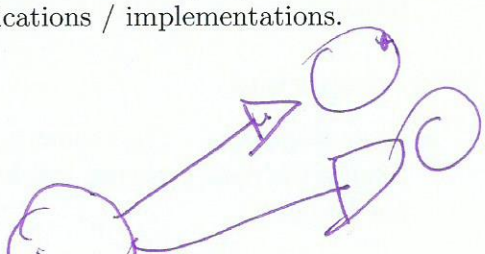
- Allow for users to view a map (divided into zones) of the restaurant.
- Overlay all the wristbands registered on the app user's name on the map to show where they are (some patrons might have multiple children to track).
- Allow for setting up of alarms and notifications based on movement of tracked wristbands.
- Allow for a user to select a restaurant, and a restaurant branch in order to get the map for the set-up.
- Enable the user to link the wristband to their mobile application

1.2.3 Web API

The system must have a central web based API that concerns itself with gathering information and supplying information from the system. This API will be used by:

- Mobile Application
- Web Administration
- A Service, if needed, that can push information to devices

Therefore this API must be loosely coupled from any implementation using it , as it needs to be re-used for multiple implementations, and possible future applications / implementations.



1.2.4 Basic Reporting

- Facility usage (zone based) by children. This will enable a restaurant to see statistics on which areas are most popular for children.
- Usage statistic of mobile app users that could possibly enable restaurant chains to implement a loyalty program.
- Health reports, reports that will indicate the health of the hardware being used. This should pick up trends that are impossible with the restaurant layout.

1.2.5 Embedded Hardware and prototyping

Physical hardware devices used in this project includes:

- Receiver/Reader prototype
  - These are devices that are placed at predetermined locations within the designated area where tracking will occur in order to pick up wristbands and their relative signal strengths. These devices will use wireless communication will to communicate this information to an access point with internet access
- Wristband prototype
  - These can be active or passive RFID tags, or wireless transponders/transceivers. They will be used to communicate its existence in range of a receiver/reader and its signal strength (RS SI)
- Electronic tracking device
  - This device, when developed, will be a piece of hardware the patron can put on their table with LED's representing zones on the restaurant map. However, for this project it is not of concern, however, it should be kept in mind during implementation for future implementation and programming.

1.3 Use case prioritization

Here we are considering a simple 3 level use case prioritization, with critical functionality (a use case or function that is absolutely essential), important functionality (if a system would still be basically functioning but still less than the client specification) and, nice-to-haves (added functionality that the client would not consider core but would add to the presentation of the project).

1.3.1 Critical

- Database - A database containing all relevant information about users, restaurants/stores and the RFID bands and readers.
- Web Administration - This requirement is essential. It allows the proper operation of the program, as web administration allows installation of this system.
- Mobile Administration - This is the part of the system the end-user will be able to use. This is one of two main interfaces with the end-user.
- WEB API - The Web API is the core of the system, it provides functionality to the client browsers and the mobile applications.

1.3.2 Important

- Basic Reporting - The reporting portion of the system is useful for logging and statistics. Even though the data gathered will be useful, the core system will still function. (Not to say this will not be done.)

1.4.2 Use Case Login / Locate Child

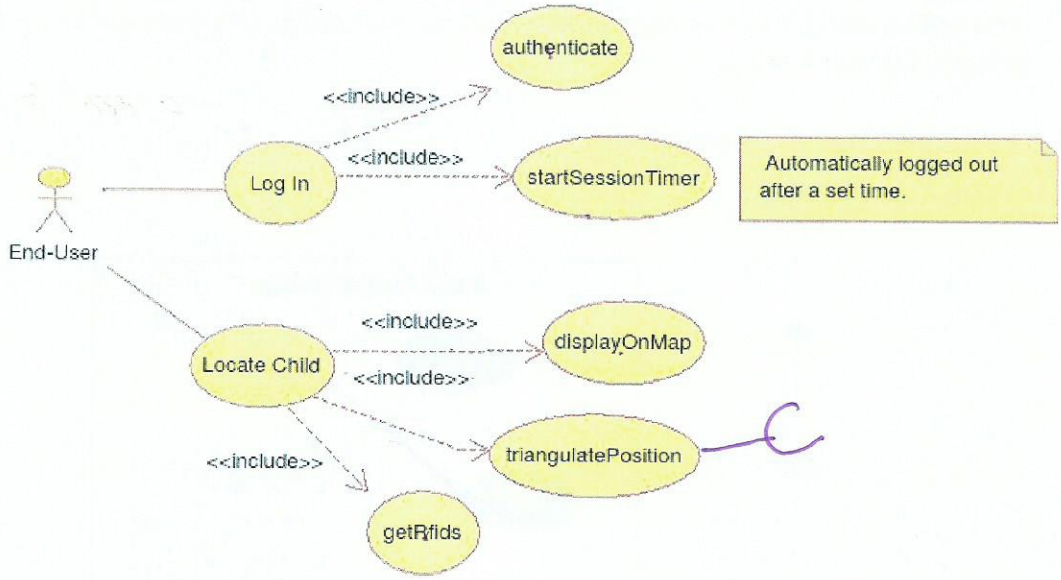


Figure 2: Use Case Diagram - Login / Locate Child

- Pre-Conditions - User needs to have their log in details already on the system
- Post-Conditions - Session timer should start to automatically log out a user after a specified amount of time. User should stay authenticated during this time.

1.4.3 Use Case Map / Link / Zones / Reporting

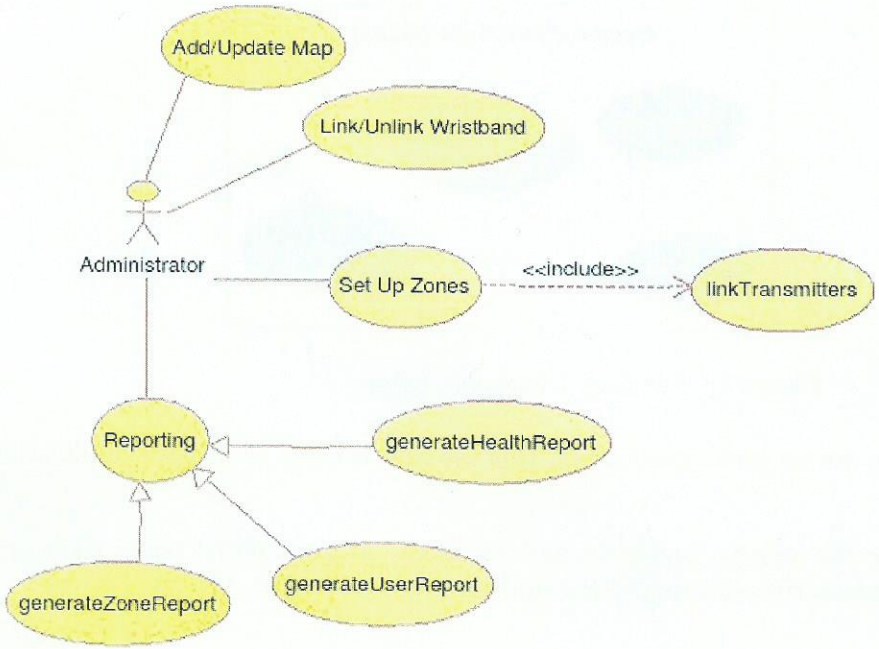


Figure 3: Use Case Diagram - Map / Link / Zones / Reporting

- Pre-Conditions - Map of the venue needs to be downloaded to the device. Wristband should not be linked to another user. Zones should be defined. Logs should be stored for reporting to work.