Client Name: DVT
Project Name: KinderFinder

Architecture Requirements

Team Name: MAU Technologies

Uteshlen Nadesan	28163304
Michael Johnston	12053300
Po-Han Chiu	11063612

https://github.com/MrBean355/KinderFinder.git

Version: 0.1

Change History:
The Summary here.

23 May 2014

Contents

1 Access Channel Requirements

The system's services are to be accessed through the following channels:

- **Web Portal**. Accessed through a web browser. This is where the system administration is done, as well as report viewing.
- **Mobile App**. Accessed on Android devices. The parents who are monitoring their children will view their location on the app.

2 Quality Requirements

There are numerous quality requirements that are quantifiable:

- **Performance.** The system needs to give real-time information of where the child is now, not a minute ago. As soon as the child leaves the area, the parents need to be notified.
- **Reliability.** It is critical that parents using the system can rely on it, because they want to be certain where their children are at all times. The system must always report the correct location of the transmitters.
- **Scalability.** It needs to be easy to add or remove hardware to/from the system, so that restaurants can expand or shrink their monitoring zones with little effort and cost.
- **Flexibility.** Different restaurants should be able to have their system customised to their needs, so they can use the system to its full potential.
- **Maintainability.** The hardware and software components need to be easily accessible so that maintenance can be performed efficiently. The web administration portal must allow the users to easily maintain the system.
- **Auditability.** The system will have a reporting module that must report statistics such as which zones are most popular and whether any devices are damaged.
- **Integrability.** The system must be easily integrable with different hardware, not locked to one model.
- **Cost.** The major cost of the system will be attributed to the hardware, such as RFID wristbands and receivers. The hardware must be affordable but decent quality, so that it can be relied upon without costing fortunes.
- **Usability.** This could be measured by means of a survey, to find out how user-friendly the users find the system.

3 Integration Requirements

The system does not need to integrate with any external systems. However, since we are focusing on the software side, the hardware can be integrated with the software in the future (possibly by us if we have the time). For this, the hardware (RFID transmitters and receivers) must be integrated into the system, replacing any dummy ”hardware” in place.

4 Architecture Constraints

The technologies that must be used are as follows:

- Microsoft SQL Server 2012 Express
- Xamarin (Android app)

- Microsoft Visual Studio 2013
- ASP.NET MVC 5 & WebAPI
- C#, HTML 5, JavaScript
- Entity Framework (ORM)

The architecture as provided by the client is as follows:

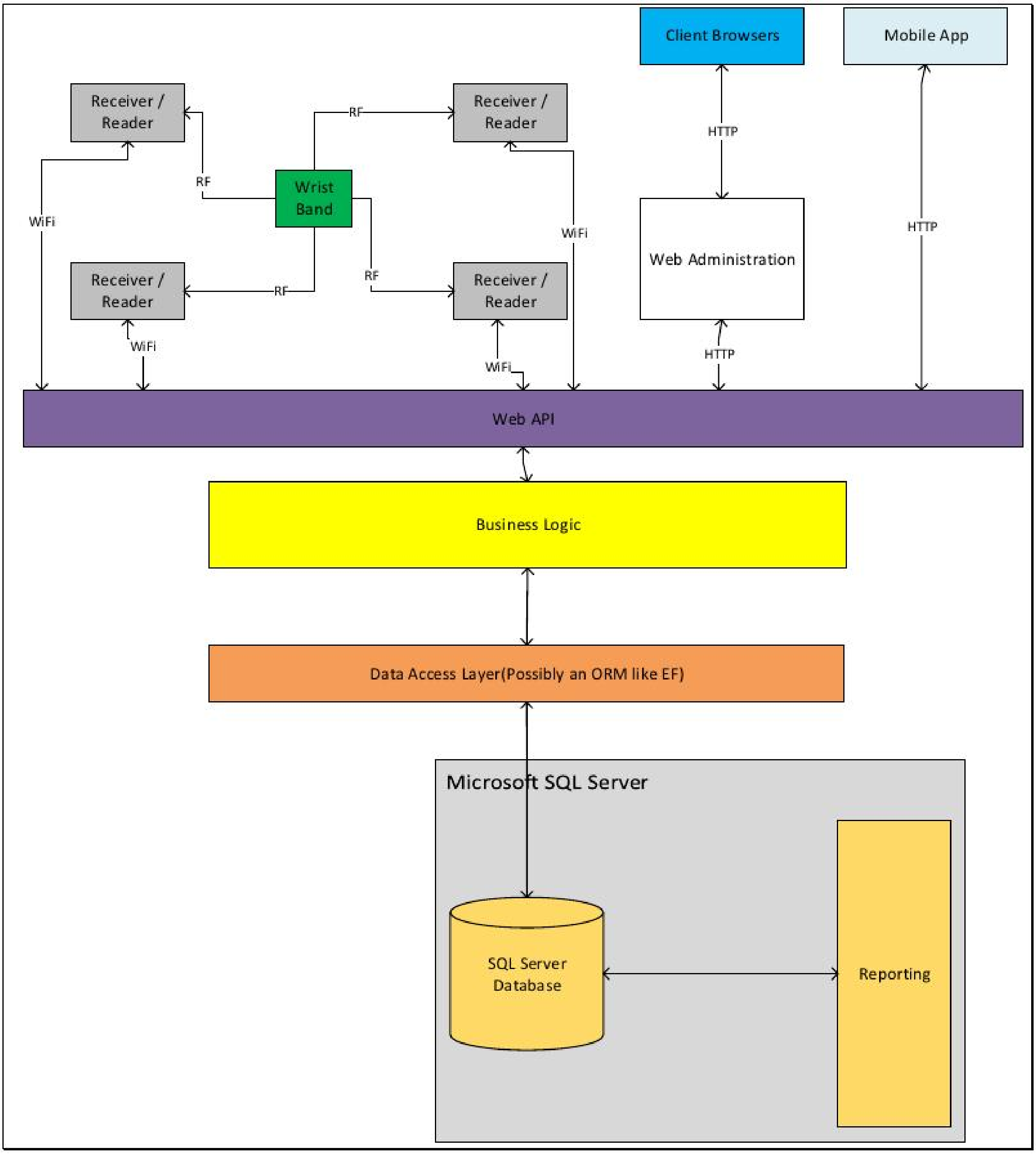

Figure 1: System architecture.