Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Fall 2021

Raytheon Team
WiDros

Daniel Tam
Joell Soriano
Joshua Pearson
Renato Cruel Amado
Tiffany Frias

Revision History

Revision	Date	Author(s)	Description
0.1	09.28.2021	JP	Document creation
0.2	09.29.2021	JP	Added diagrams for system/subsystems
0.3	10.21.2021	DT	Revision From AWS to Azure Cosmos DB API for MongoDB subsystem
0.4	10.24.2021	DT, JP, JS, RCA, TF	Review for final draft
0.5	10.25.2021	JP	Final draft completed
0.6	11.5.2021	JS	Revisions on Database to reflect use of Daniels SQL DB

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

WiDROS is a system that allows users to see real-world coverage range of all wireless network present at a building in the University of Texas at Arlington, through an augmented reality environment. Users of WiDROS software will be able to experience an augmented reality environment which will display wireless networks and relevant information. They will also be able to see the coverage range of all wireless network present on the building through this augmented reality environment.

2 SYSTEM OVERVIEW

The system accepts real-world wireless signal data collected by a Raspberry Pi placed on a drone. The signal information was meant to stored on Azure Cosmos DB API for MongoDB then processed and packaged using Unity into visual domes that indicate the coverage range of each network. Due to complications and shifts of the direction of the project we implemented a home grown SQL database instead. We intend on still using one of the mentioned prior if time allows. Our product will map these visual domes on to an accurate augmented reality environment of a building in the University of Texas at Arlington. The product will allow users use a Microsoft HoloLens to experience the altered reality environment of the building along with the network coverage domes. It can be used to accurately view the coverage of each wireless signal on the building and optimize router placement to ensure better coverage.

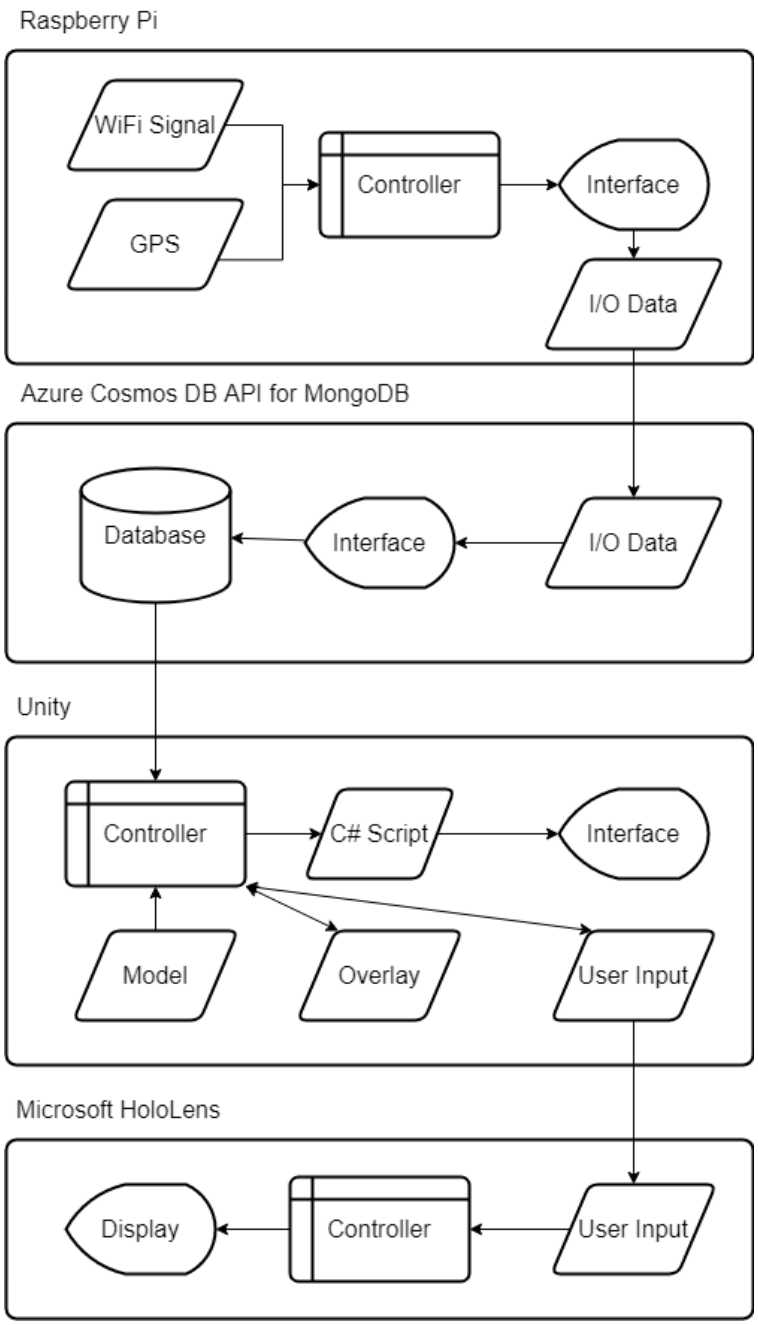

Figure 1: System Architecture

3 Raspberry Pi Subsystems

The Raspberry Pi unit that is equipped with sensors to collect data from the real world. This unit will collect WiFi signals and GPS coordinates to assist in obtaining information about the wireless signal and its approximate location relevant to the drone delivery system. The information will be compiled and delivered later to an Azure Cosmos DB API for MongoDB database.Once again we were unable to implement do to time and budget constraints so we went with a home made database instead.

3.1 Layer Hardware

The Raspberry Pi has several sensors that will detect the following properties: WiFi SSID, WiFi Strength and global position.

3.2 Layer Operating System

The Raspberry Pi uses the Kali linux distribution of the Raspberry Pi OS.

3.3 Subsystem 1

The Data File subsystem will host all the data processed by the Raspberry Pi sensors. The data file will also contain data for heavier calculations that may be necessary to create the augmented reality simulation and triangulation of WiFi domes, such as distance of signal in order to better triangulate.

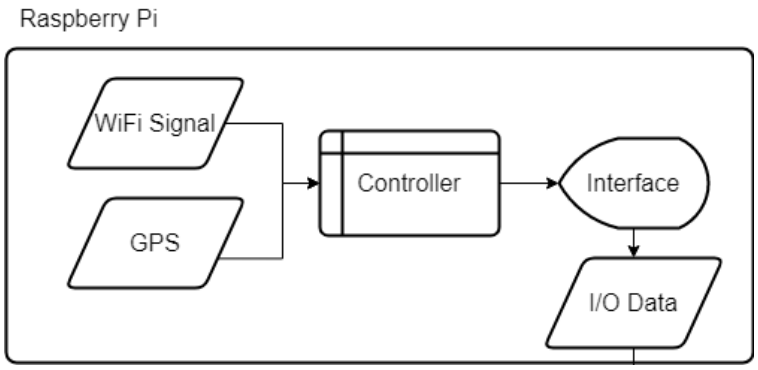

Figure 2: Raspberry Pi data flow

3.3.1 Subsystem Hardware

The hardware required is a Raspberry Pi, GPS antenna and MicroSD card. There is also the drone component for transporting the Raspberry Pi for data collection.

3.3.2 Subsystem Software Dependencies

The controller component is the main set of scripts on the Raspberry Pi that will process the sensor data and post the data files. The Pi contains multiple python scripts that are able to read, process, and post the data collected by the sensors. The controller will send data files to the Azure Cosmos DB API for MongoDB database.

3.3.3 Subsystem Programming Languages

This subsystem uses Python, because it gave us great versatility in case any shifts in the scope of the project changed. We determined it was probably best suited for most things.

4 SQL Database Subsystems

The Azure Cosmos DB API for MongoDB makes it easy to use Cosmos DB as if it were a MongoDB database. Azure Cosmos DB API for MongoDB implements the wire protocol for MongoDB. This implementation allows transparent compatibility with native MongoDB client SDKs, drivers, and tools. Due to time and budget constraints we went with a Daniels home made sql database which was much easier to implement because of its simplicity, and we did not have any concerns on getting it approved by the sponsor or have to worry about monetary issues.

4.1 Layer Operating System

The SQL database was capable of being used with a variety of OS systems and did not limit us to anything specific.

4.2 Layer Software Dependencies

Our SQL database does not have any software dependencies. Json files are used to store the data collected from the Raspberry Pi,which will then be uploaded to the database.

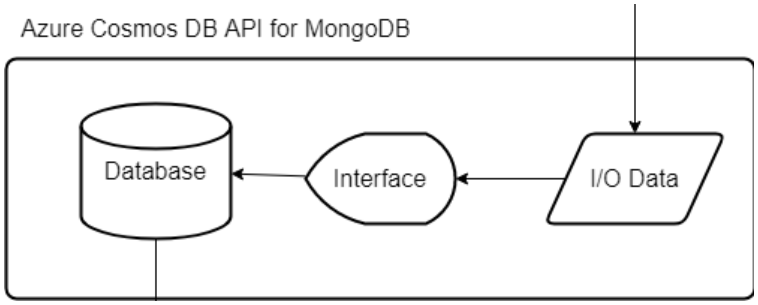

Figure 3: Example subsystem description diagram

5 Microsoft HoloLens Subsystems

The Microsoft HoloLens subsystem contains an API to interact with and retrieve data from Unity that will process the data contained within the database, and create the environment of the augmented reality simulation, an overlay that will visualize fetched data, a user interface, and a layer for user input to interact with display.

5.1 Layer Hardware

We are utilizing a Microsoft HoloLens 2, provided by the school.

5.2 Layer Operating System

Windows 10 The Microsoft HoloLens 2 runs on Windows OS.

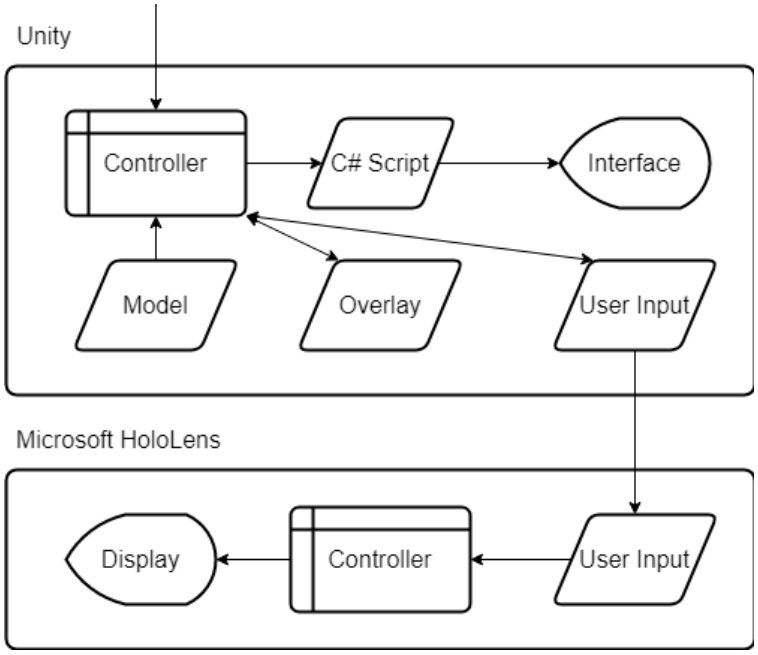

Figure 4: Data flow from Unity into the Microsoft HoloLens

References