CSCI E-97 Software Design 9/15/2020

Smart City Design Document

Date: 9/15/2020

Author: Loi Cheng

Reviewer(s):

Introduction

The Smart City Software System will fully automate the city of the future. Smart City allows city

administrators to fully automate a city through AI-powered Internet of Things (IoT) devices,

including cameras, microphones, robots, and other devices. Sensors monitor the location of

persons within the city. Robot Assistants help residents and visitors, clean the city, and respond

to emergencies. Some of the devices are controlled automatically, and all devices are able to

interact with people through a voice interface.

This design specifies the implementation of a Smart City Service. The Smart City Model Service is responsible for managing the domain entities of the Smart City System. Domain entities include the city, people, and IoT devices, including street signs, information kiosks, street lights, parking spaces, public servant robots, and vehicles. The Smart City Model Service provides an API for interacting with those objects. The API supports querying the state of the entities, as well as updating the state.

Overview

Overview of the problem to be solved. What is the problem and why is it being solved? How

will the resulting solution provide business value?

Consider adding a diagram that explains how this component fits into the overall System with

some descriptive text explaining the diagram.

Requirements

This section provides a summary of the requirements for the <Component Name>.

Provide your understanding of the requirements, both functional and nonfunctional. Reference

the provided Requirements and System Architecture documents. Do not cut and paste from the

requirements document.

The Product Manager and others can read this to understand what requirements your design

will support. There is already a requirements doc, so keep this brief and to the point,

highlighting the important requirements that the design is addressing. Structure in a way to

provide a requirements checklist for your design.

1

CSCI E-97 Software Design 9/15/2020

Use Cases

Enumerate the use cases supported by the design,

This design supports the following use cases:

Include a UML Use Case Diagram.

Include descriptions of each of the actors and use cases.

Implementation

This section of the document will describe the implementation details for ...

The implementation section should cover the following topics:

● What are the classes, and their properties, associations, and methods?

● What are the important interfaces and how they will be implemented?

● How are the requirements addressed?

How does this module fit into the overall architecture? Include a UML Component Diagram to

explain.

Class Diagram

The following class diagram defines the classes defined in this design. Remember to include

exception classes.

UML CLASS DIAGRAM GOES HERE

Class Dictionary

This section specifies the class dictionary for the class … defined within the package …

CLASS 1

Class 1 description

Methods

Method Name Signature Description

2

CSCI E-97 Software Design 9/15/2020

Properties

Property Name Type Description

Associations

Association

Name

Type Description

CLASS 2

...

Implementation Details

Explain the details of the implementation.

How do the various parts fit together or interact?

How does the design address the requirements? Justify your design decisions and how they

address the requirements.

Some implementation details may be addressed in the class dictionary, but for things that are

not, describe them here.

Remember to reference the requirements from the body of the design document to show how

your design is addressing the requirements.

Add UML Sequence Diagrams or Activity Diagrams to explain important workflows.

Exception Handling

Provide details on your exception handling. What types of exceptions are expected and how

are they handled by the design? Describe your exception classes and their properties.

Testing

Provide a testing strategy for testing the component.

● Functional

● Performance

● Regression

3

CSCI E-97 Software Design 9/15/2020

● Exception Handling

Risks

Document any risks identified during the design process.

Are there parts of the design that may not work or need to be implemented with special care or

additional testing?

4