

IAS Group 7 : Team 2
Team Requirement Document
Application Management, UI, Notification manager & Sensor
Data Binder

Param Pujara : 2020202008
Nisarg Pethani : 2020202012
Karan Bhut : 2020202015

Contents

1	Functional Overview	3
2	Block Diagram	4
2.1	Block Diagram of Subsystem	4
2.2	Application development Model Diagram & Flow with respect to subsystem	4
3	Subsystems	5
3.1	Application Manager	5
3.2	Notification Manager	5
3.3	Sensor data binding	5
3.4	User interface	5
3.5	Azure Database Handling	5
3.6	Azure Repository	5
3.7	Application	5
4	Services	6
4.1	Application Manager:	6
4.1.1	Authentication Module	6
4.1.2	Validation	6
4.2	Sensor Data binding	6
4.3	Notification Manager	6
4.4	User interface	6
4.5	Azure MySQL Handling	6
4.6	Azure Repository	7
4.7	Application	7
4.7.1	Sample Application 1	7
5	Interactions with other parts	8
5.1	Application manager with Scheduler	8
5.2	Application manager with Monitoring and Fault tolerance	8
5.3	Application manager with Sensor Manager	8
5.4	Application manager with Controller Manager	8

1 Functional Overview

Our module will take configuration files from Application developers and Platform Configurers through UI. We will validate structure of zip files through validation module. We will provide Authentication through UI. Users can login and deploy application they wanted, they can also see the notifications for application.

We are also maintaining Azure MySQL database and Azure Repository to store whole project's data in cloud. We have also built Sample Application 1 which provides transport facility using buses.

2 Block Diagram

2.1 Block Diagram of Subsystem

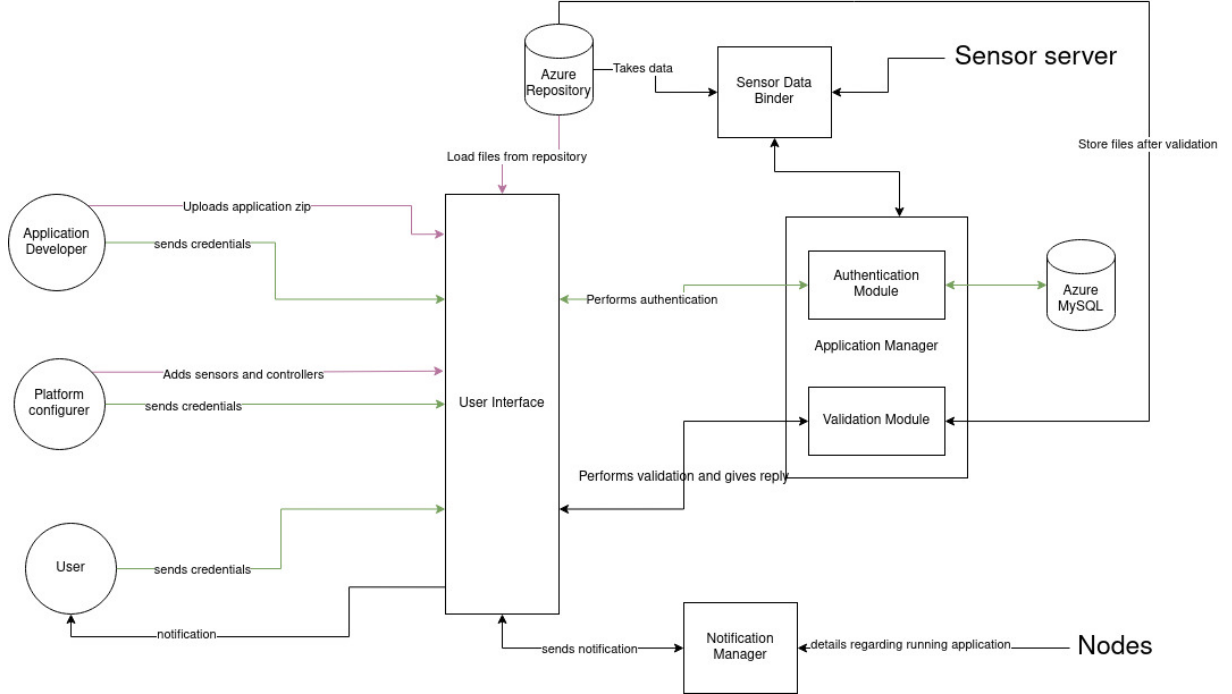


Figure 1: Block Diagram

2.2 Application development Model Diagram & Flow with respect to subsystem

1. Authentication module will verify credentials of all actors.
2. Platform configurer will add sensors and controllers instances to the platform.
3. Application manager will validates format and structure of different files uploaded.
4. Application developer will upload required sensors information, application code to the platform.
5. Application Manager will validate them and stores files on Azure Repository.
6. User will enter his/her preferences where and when to deploy the application.
7. Application manager will send those details further to deploy application.
8. At the end Application will be deployed on nodes and Notifications from application will be sent to notification page, from where user can read them.

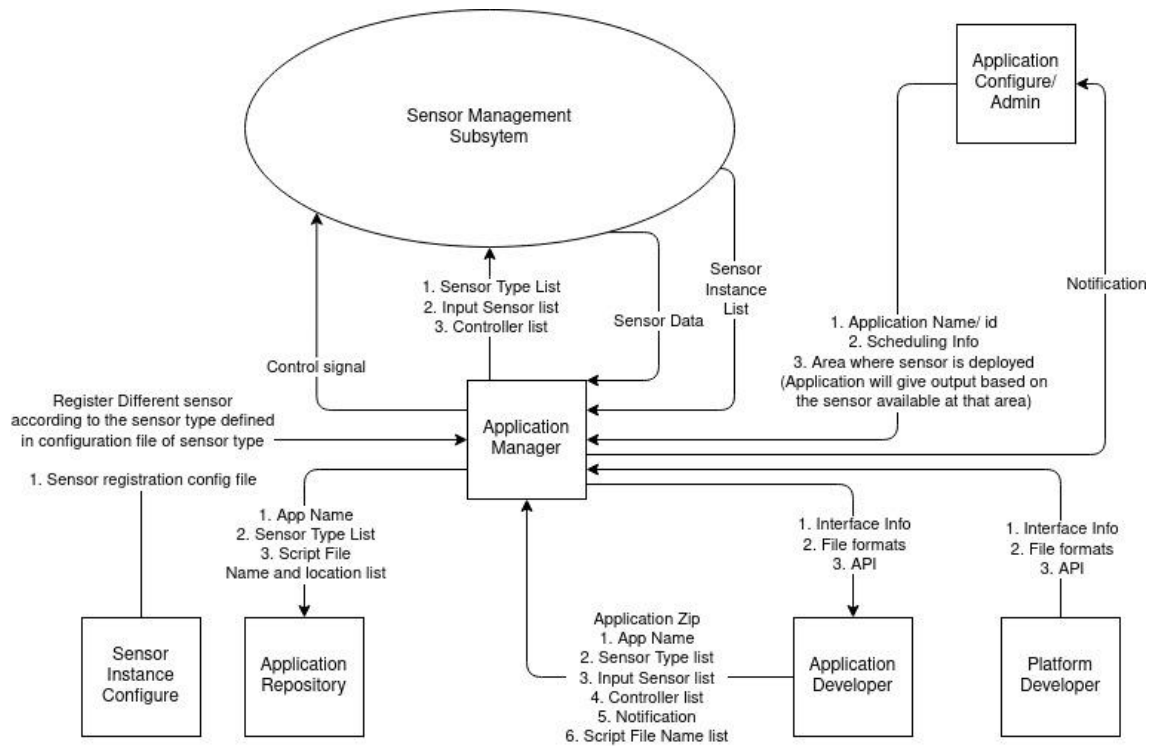


Figure 2: Application Development Diagram

3 Subsystems

3.1 Application Manager

- Authentication module
- Validation module

3.2 Notification Manager

3.3 Sensor data binding

3.4 User interface

3.5 Azure Database Handling

3.6 Azure Repository

3.7 Application

4 Services

4.1 Application Manager:

4.1.1 Authentication Module

- Authenticates Application developers and Users through their passwords. It also authenticates Platform Configurer with his password and Secret Key.

4.1.2 Validation

- Validates structure of configuration zips submitted by platform configurer and application developer.

4.2 Sensor Data binding

- This module will bind sensor instance with index mentioned in application zip.
- User will provide location information of sensor instance and application developer would have mentioned Sensor type information in the App.
- This module will search a sensor info and sensor type info into database and select one of those sensor.
- Selected sensor will be put into non useful for that particular app.
- So, Same sensor will not be selected again in same application.
- Sensor data binder will binds all sensors of an application in this manner.
- It will reply success message or error if any happens to be occurred.
- The same binding process will be followed for controller data binding by this module.
- This is how sensor and controller data binding happens.

4.3 Notification Manager

- It will display notifications of different Applications executed by users.

4.4 User interface

- It includes Registration and Login pages for platform configurer, application developer and users.
- It also includes facility for platform configurer to upload different ZIP files like sensor_type, sensor_instance, controller_type and controller_instance.
- Application developer can upload application.zip file containing application requirements and code.
- user can deploy Application through it. User can also see different notifications of applications previously deployed.

4.5 Azure MySQL Handling

- We have managed Azure MySQL database for whole Project.
- It includes tables to store data of different sub-systems like different sensors information, application binding information, nodes information, etc.

4.6 Azure Repository

- We have managed Azure Repository for entire project to store codes for different module, configuration files for Kafka and Docker and json files for sensors and controllers. So that we can access them from distributed locations remotely.

4.7 Application

4.7.1 Sample Application 1

- Application is built to provide transport facility using buses.
- Sensors:
 - GPS: Sends placeholder-id co-ordinates. Each bus is installed with one such sensor, one sensor is at IIIT-H campus and each police barricade has one such sensor.
 - Biometric: Sends placeholder-id person-id (it will be unique on place-id level)
 - Temperature: Sends current temperature of the place
 - Light Sensors: Sends Lux Level of the place
- Controllers: Lights, ACs, Buzzers
- Cases:
 - Whenever someone boards a bus, he/she will do biometric check-in. Fare will be calculated based on distance multiplied by fixed rate that information (Bus-id, Passenger-id and Fare) will be sent to guard.
 - If temperature is more than a threshold switch on the air conditions or lighting is lower than a lux level switch on the lights.
 - If more than two (≥ 3) buses come in a circle of given radius, except one send buzzer command to the rest. Run this service after every 1 minute.
 - Also, Whenever a bus comes closer to a barricade by a threshold distance start/trigger an algorithm which sends an email to administration mentioning unique identifier of the bus an email body containing information of bus and notifying them.

5 Interactions with other parts

5.1 Application manager with Scheduler

Application Manager will provide Scheduler with the application and the meta file of the application, where information related to scheduling are present (start and end times, job type etc)

5.2 Application manager with Monitoring and Fault tolerance

- The monitoring and fault tolerance module interacts with Application manager, Authentication Manager and Analytic module.
- For all these modules, periodic monitoring will be provided through heart beat message.

5.3 Application manager with Sensor Manager

Sensor manager will provide sensor's information to the application manager which will be stored at application database along with other application information and passed to scheduler in form of meta data with application's other information.

5.4 Application manager with Controller Manager

Control manager will send Controller's information to the application manager and that information will be stored at application database along with the other app information.