

SE 4352.001 Software Architecture

Final Term Project

# **Pig Farmer Automated Resort Software (PFARS)**

By Alex Lundin, Geet Babaria, Justin Horak

# Abstract

Pig Farmers Automated Resort Software (PFARS) system is a fully integrable resort software system built to client specifications. PFARS limits the need for actual physical employees by introducing Artificial Intelligence into daily operations.

The PFARS based resort is an automated vacation hotspot that allows for clients to do as little as possible in their normal daily needs and tasks. This includes but is not limited to such things as preference control, in which customer prior to completing their check-in, set their temperature, music, and snack preferences. Additionally, upon completion a RFID card is sent to the customer with a set range of check-in to check-out times. This card can be reactivated for additional resort visits.

Each resort customer sets their initial preferences, followed by a PFARS request for permission to establish connection with social media sites to specify more preferences such as uploading pictures of family on to digital picture frames. At this point the check in is complete and the preferences are sent to the PFARS server.

Once, PFARS sends a request for check-in to the server, staff will be alerted, and the appropriate requested items are collected and set into the customers temporary domicile. PFARS based temporary domiciles, are based on a centralized resort server, in which the resort staff monitors.

Staff members are required to have security permissions based on job title which will limit access according to their clearance. Customers will also have special permissions based on their needs and additional service costs.

Each domicile is smart appliance ready and adaptable to additional hardware. The air conditioning, refrigerator, domicile door locks and security cameras will be minimal requirements for functionality when automating PFARS.

Optional customer functions are as follows: voice activated system control, scheduling and reminder system, and automated concierge service which is integrated into the PFARS customer permissions upon preference selection.

Automated concierge service will be connected to the local community which will allow for item delivery or reservation services for events and/or restaurants.

NOTE: Preferences can be changed at any time via PFARS resort app, and if selected can also be controlled through the voice activated system control.

## Revision History

Date	Version	Description	Author
11/20/2018	1.0	Document Creation	Geet
11/21/2018	1.1	Decided on content	Alex & Geet
11/21/2018	1.2	Edited heading and list levels	Alex
11/21/2018	1.3	Pasted information from Phase I	Alex
11/21/2018	1.4	Formatted everything consistently	Alex
11/23/2018	1.5	Filled out Introduction	Alex
11/25/2018	1.6	Group Workday (Section 2,3,4)	Alex, Geet, Justin
11/27/2018	1.7	Finish Diagrams & Documentation	Alex, Geet, Justin

1. Introduction .....	7
1.1. Purpose .....	7
1.2. Scope .....	7
1.2.1. Scope Statement .....	7
1.3. Product Description .....	7
1.4. Definitions, acronyms & abbreviation .....	8
1.4.1. PFARS - Pig Farmers Automated Resort Software .....	8
1.4.2. Domicile - home dwelling for the resort customers .....	8
1.5. Summary .....	8
1.6. Overview .....	8
2. Overall description .....	9
2.1. Product perspective .....	9
2.3. User interfaces .....	11
2.3.1. Device: Computer .....	11
2.3.1.1. Web User Interface .....	11
2.3.1.1.1. Action Methods .....	11
2.3.1.1.1.1. Point-and-Click .....	11
2.3.1.1.1.2. Touch-Screen (if applicable) .....	11
2.3.1.1.1.3. Standard-Keybaord .....	11
2.3.2. Device: Tablet .....	11
2.3.2.1. Resort Application .....	11
2.3.2.1.1. Action Methods .....	11
2.3.2.1.1.1. Voice Command .....	11
2.3.2.1.1.2. Virtual Keyboard .....	11
2.3.2.1.1.3. Keyboard (if Attached) .....	11
2.3.2.1.1.4. Touch-Screen .....	11
2.3.3. Device: Phone .....	11
2.3.3.1. Resort Application .....	11
2.3.3.1.1. Action Methods .....	11
2.3.3.1.1.1. Voice Command .....	11
2.3.3.1.1.2. Virtual Keyboard .....	11

2.3.3.1.1.3.	Keyboard (if Attached) .....	11
2.3.3.1.1.4.	Touch-Screen .....	11
2.4.	Hardware .....	11
2.5.	Software .....	12
2.6.	Operations.....	12
2.7.	Site adaptation requirements .....	<b>Error! Bookmark not defined.</b>
2.8.	Product functions .....	12
2.9.	User characteristics.....	14
2.10.	Constraints.....	14
2.11.	Assumptions and dependencies .....	14
3.	Architectural Representation .....	15
3.1.	Representation.....	15
3.2.	Views .....	15
4.	Specific Requirements .....	18
4.1.	External.....	18
4.2.	Specific requirements.....	19
4.3.	Performance requirements.....	19
4.3.1.	Network Availability 100%.....	19
4.3.2.	Response time, Highest priority for customers, lowest priority for inhouse maintenance and records.....	19
4.3.3.	RFID response time needs to be within 2 seconds.....	19
4.3.4.	Voice Recognition need to be configured in 1 min.....	19
4.3.5.	Voice recognition also acts as secondary key .....	19
4.4.	Software system attributes.....	19
4.5.	Other requirements .....	21
4.5.1.	System Evaluation .....	21
4.5.1.1.	Correctness .....	21
4.5.1.1.1.	Does the system reflect the requirements of the architectural document	21
4.5.1.2.	Efficiency .....	21
4.5.1.2.1.	Resource usage and software functionality.....	21
4.5.1.3.	Flexibility.....	21

4.5.1.3.1. IoT expansion in domiciles .....	21
4.5.1.4. Integrity/ Security .....	21
4.5.1.4.1. Prevention against attacks to the resort system .....	21
4.5.1.5. Interoperability .....	21
4.5.1.5.1. Software and hardware integration.....	21
4.5.1.6. Maintainability .....	21
4.5.1.6.1. Ease of maintenance for non-technical employees .....	21
4.5.1.7. Reliability .....	21
4.5.1.7.1. Ensures system backups and alternative resources to continue servers in a limited environment.....	21
4.5.1.8. Reusability .....	21
4.5.1.8.1. Able to adapt to new technologies and repeated use.....	21
4.5.1.9. Testability .....	21
4.5.1.9.1. Able to meet requirements put forth by the client that also reflect in the architectural documentations .....	21
4.5.1.10. Usability.....	21
4.5.1.10.1. Employees and resort clients can with minimum learning operate their respected role-based system views .....	21
4.5.1.11. Availability .....	21
4.5.1.11.1. System availability to client, employees, and resort clients.....	21
References.....	22

# 1. Introduction

## 1.1. Purpose

### 1.1.1. Purpose Statement

This document encapsulates the entire set of requirements gathered for PFARS during the Fall 2018 semester of Software Architecture and Design.

### 1.1.2. Audience

The Audience includes Team 9, Teaching Assistant Ms. Munawara Munia and Dr. Pushpa Kumar.

## 1.2. Scope

### 1.2.1. Scope Statement

The scope of this document stems from the requirements gathering session during phase I of the project. This document evolves directly from the initial set of requirements and grows with our understanding of Software Architecture as the semester progresses.

## 1.3. Product Description

### 1.3.1. Product Name

#### 1.3.1.1. PFARS

### 1.3.2. Product shall:

1.3.2.1. Provide a set of automated domiciles for a wealthy customer

1.3.2.2. Tailor the digital experience of the domicile, to the customer

1.3.2.3. Tie in to surrounding services, like restaurant and stores

1.3.2.4. Prioritize security of the customer with:

1.3.2.4.1. Authorization for access to any sensitive data

1.3.2.4.2. Data encryption for all user data transfer

1.3.2.4.3. Physical locks on the domiciles

1.3.2.5. Prioritize a modifiable design

1.3.2.6. Prioritize good user experience

### 1.3.3. Product shall not:

1.3.3.1. Include on campus staff for any types of external surrounding services

1.3.3.1.1. Example, there will be no chef on staff since the software ties into nearby restaurants

### 1.3.4. Application of software

The software will allow the system's owner to go to market with a brand-new vacation option for the vacation business.

### 1.3.5. Relevant benefits

1.3.5.1. Access to any close proximity good and services

- 1.3.5.2. All-inclusive vacation getaway with amenities on site
    - 1.3.5.3. Unique user experience
  - 1.3.6. Objectives and goals
    - 1.3.6.1. Factor security into all design decisions
    - 1.3.6.2. Design modifiable architecture
    - 1.3.6.3. Design smart home tie in
- 1.4. Definitions, acronyms & abbreviation
  - 1.4.1. PFARS - Pig Farmers Automated Resort Software
  - 1.4.2. Domicile - home dwelling for the resort customers
- 1.5. Summary
 

This document describes the evolution of requirements to the final deliverable for Architecture and Design Fall 2018. The first presentation from our Stakeholder, Gordon Arnold, gave us an opportunity to elicit what the system must do. During phase I of the project we documented the functional and non-functional requirements. During phase II, we refined the functional and non-functional items and created a new Software Architecture Document. The SAD further extended the representation of the initial requirements into a more comprehensive format the gives a holistic representation of the final deliverable.
- 1.6. Overview
  - 1.6.1. The remainder of this document will break down the stakeholder needs to atomic requirements that can be implemented during system design. Some requirements will include high level views so architects can design the software to specification.
  - 1.6.2. Organization
    - 1.6.2.1. Overall Description
    - 1.6.2.2. Architectural Representation
    - 1.6.2.3. Views
    - 1.6.2.4. Specific Requirements



## 2. Overall description

Describe general factors that affect the product and requirements. This section provides background information that is necessary to define requirements in section 3.

### 2.1. Product perspective

- 2.1.1. PFARS is a standard web-based architecture that uses a python-based driver, css, and html5 for cross-platform support. PFARS utilizes a web application source for both Android and iOS applications.

### 2.2. System interfaces

- 2.2.1. PFARS uses cascading style sheets (CSS) to adjust the format for each class of devices

- 2.2.1.1. Device Class

- 2.2.1.1.1. Personal computer
- 2.2.1.1.2. Tablets
- 2.2.1.1.3. Cell phone

- 2.2.2. Class Optimizations

- 2.2.2.1. Personal computer

- 2.2.2.1.1. Specifications

- 2.2.2.1.1.1. CSS Screen Size

- 2.2.2.1.1.1.1. @media only screen and (min-width : 1224px)

- 2.2.2.1.1.2. User View

- 2.2.2.1.1.2.1. Full Dashboard Access
- 2.2.2.1.1.2.2. User Settings and Appearance
- 2.2.2.1.1.2.3. Data input
- 2.2.2.1.1.2.4. Media upload
- 2.2.2.1.1.2.5. Resort Background Video
- 2.2.2.1.1.2.6. Full Application Usage

- 2.2.2.2. Tablets

- 2.2.2.2.1. Specifications

- 2.2.2.2.1.1. CSS Screen Size

- 2.2.2.2.1.1.1. @media only screen and (min-device-width : 768px) and (max-device-width : 1024px)

2.2.2.2.1.2.	User View
2.2.2.2.1.2.1.	Condensed Dashboard
2.2.2.2.1.2.2.	Data input
2.2.2.2.1.2.3.	Media upload from device
2.2.2.2.1.2.4.	Resort Logo with Background Image
2.2.2.2.1.2.5.	Calendar
2.2.2.2.1.2.6.	Alarm Setup
2.2.2.2.1.2.7.	Full Application Usage
2.2.2.3.	Phones
2.2.2.3.1.	Specifications
2.2.2.3.1.1.	CSS Screen Size
2.2.2.3.1.1.1.	@media only screen and (min-device-width : 320px) and (max-device-width : 480px) and (orientation : landscape) and (-webkit-min-device-pixel-ratio : 2)
2.2.2.3.1.2.	User View
2.2.2.3.1.2.1.	Limited Dashboard
2.2.2.3.1.2.2.	Change Room Settings
2.2.2.3.1.2.3.	Media Upload from Device
2.2.2.3.1.2.4.	Resort Logo
2.2.2.3.1.2.5.	Calendar
2.2.2.3.1.2.6.	Alarm Setup

- 2.3. User interfaces
  - 2.3.1. Device: Computer
    - 2.3.1.1. Web User Interface
      - 2.3.1.1.1. Action Methods
        - 2.3.1.1.1.1. Point-and-Click
        - 2.3.1.1.1.2. Touch-Screen (if applicable)
        - 2.3.1.1.1.3. Standard-KeyBoard
  - 2.3.2. Device: Tablet
    - 2.3.2.1. Resort Application
      - 2.3.2.1.1. Action Methods
        - 2.3.2.1.1.1. Voice Command
        - 2.3.2.1.1.2. Virtual Keyboard
        - 2.3.2.1.1.3. Keyboard (if Attached)
        - 2.3.2.1.1.4. Touch-Screen
  - 2.3.3. Device: Phone
    - 2.3.3.1. Resort Application
      - 2.3.3.1.1. Action Methods
        - 2.3.3.1.1.1. Voice Command
        - 2.3.3.1.1.2. Virtual Keyboard
        - 2.3.3.1.1.3. Keyboard (if Attached)
        - 2.3.3.1.1.4. Touch-Screen
- 2.4. Hardware
  - 2.4.1. Smart Devices
    - 2.4.1.1. Sensors
    - 2.4.1.2. Cameras
    - 2.4.1.3. Smart Locks
    - 2.4.1.4. RFID scanners
    - 2.4.1.5. Microphone
  - 2.4.2. Domicile Hub
    - 2.4.2.1. Raspberry pi v3
    - 2.4.2.2. Retroflag Raspberry pi v3 case
  - 2.4.3. Centralized Hub
    - 2.4.3.1. 2U 4Node 24 Bay Xeon Scalable

## 2.5. Software

### 2.5.1. Domoticz

2.5.1.1. Use: Linux Integration for Automation Tools Software

2.5.1.2. Version: 4.9700

2.5.1.3. Source: <https://domoticz.com/downloads/>

### 2.5.2. Epiphany

2.5.2.1. Use: Web to Application conversion Software

2.5.2.2. Version: 3.29.3

2.5.2.3. Source: <https://www.ubuntu.com/downloads>

### 2.5.3. Ubuntu Server

2.5.3.1. Use: Server Software

2.5.3.2. Version 18.04 LTS

2.5.3.3. Source: <https://www.ubuntu.com/downloads>

### 2.5.4. Python

2.5.4.1. Use: Driver Software Language

2.5.4.2. Version: 3.7.1

2.5.4.3. Source: <https://www.python.org/downloads/>

### 2.5.5. MySQL

2.5.5.1. Use: Database Software

2.5.5.2. Version: 8.0.13

2.5.5.3. Source: <https://dev.mysql.com/downloads/>

\*\*\*\*Note: All software is open-source mitigating overall system costs\*\*\*\*

## 2.6. Operations

### 2.6.1. Specify normal and special operations required by each user such as:

2.6.1.1. Modes of operations the user can initiate

2.6.1.2. Periods of user activity followed by inactivity

2.6.1.3. Data processing support functions

2.6.1.4. Backup and recovery operations test

## 2.7. Product functions

### 2.7.1. Administrative function

- 2.7.1.1. Registration for:
  - 2.7.1.1.1. Customer
  - 2.7.1.1.2. Employee
  - 2.7.1.1.3. Vendor
- 2.7.2. Amenities
  - 2.7.2.1. Indoor skydiving
  - 2.7.2.2. Gym
  - 2.7.2.3. Pool
  - 2.7.2.4. Rock wall
- 2.7.3. Audit Tracking
  - 2.7.3.1. Check-In Time
  - 2.7.3.2. Check-Out Time
  - 2.7.3.3. Inventory of items in domicile
- 2.7.4. Authentication
  - 2.7.4.1. RFID
  - 2.7.4.2. Gate to community
  - 2.7.4.3. Door to house
  - 2.7.4.4. Activity centers
  - 2.7.4.5. Pets also have RFID
- 2.7.5. Authorization levels
  - 2.7.5.1. Classification of users
    - 2.7.5.1.1. Customers
      - 2.7.5.1.1.1. Level 1
      - 2.7.5.1.1.2. Level 2
      - 2.7.5.1.1.3. Level 3
    - 2.7.5.1.2. Parents/Child
      - 2.7.5.1.2.1. Parental control option for parents of children
    - 2.7.5.1.3. Employee
      - 2.7.5.1.3.1. Admin
      - 2.7.5.1.3.2. User
    - 2.7.5.1.4. Vendor
      - 2.7.5.1.4.1. User
- 2.7.6. Backup system
  - 2.7.6.1. Parsed down version of full system
  - 2.7.6.2. Only includes life critical functionality
- 2.7.7. External Interfaces
  - 2.7.7.1. Plugins to all external services
    - 2.7.7.1.1. Food
    - 2.7.7.1.2. Transport
- 2.7.8. Historical Data
  - 2.7.8.1. Customer choices
  - 2.7.8.2. Customer preferences
- 2.7.9. Legal or Regulatory Requirements

- 2.7.9.1. Access restrictions
      - 2.7.9.1.1. No access to property after checkout
    - 2.7.9.2. Camera monitoring
      - 2.7.9.2.1. Privacy policy
      - 2.7.9.2.2. Waiver signatures
  - 2.7.10. Reporting Requirements
    - 2.7.10.1. History of customer choices
    - 2.7.10.2. History of customer preferences
  - 2.7.11. Smart Home
    - 2.7.11.1. Digital picture frames
      - 2.7.11.1.1. Uploaded from customer's social media
  - 2.7.12. Transaction corrections, adjustments and cancellations
    - 2.7.12.1. Items ordered to domicile
- 2.8. User characteristics
  - 2.8.1. User may or may not be familiar with technology, but able to follow simple instruction of using device. User may or may not be familiar with smart devices.
- 2.9. Constraints
  - 2.9.1. Client-budget constraint, physical hardware must be purchased
  - 2.9.2. Knowledge constraint, limitations based on the knowledge of the architect, and development time in the areas of IoT
  - 2.9.3. Time constraints, client has to create revenue as soon as possible, development time may or may not be affected
  - 2.9.4. Quality-control constraints, knowledge and time constraints may prevent testing and better development methods
- 2.10. Assumptions and dependencies
  - 2.10.1. Individuals and employees will have different access rights
  - 2.10.2. Every individual whether client or employee will wear a RFID tag on resort premises

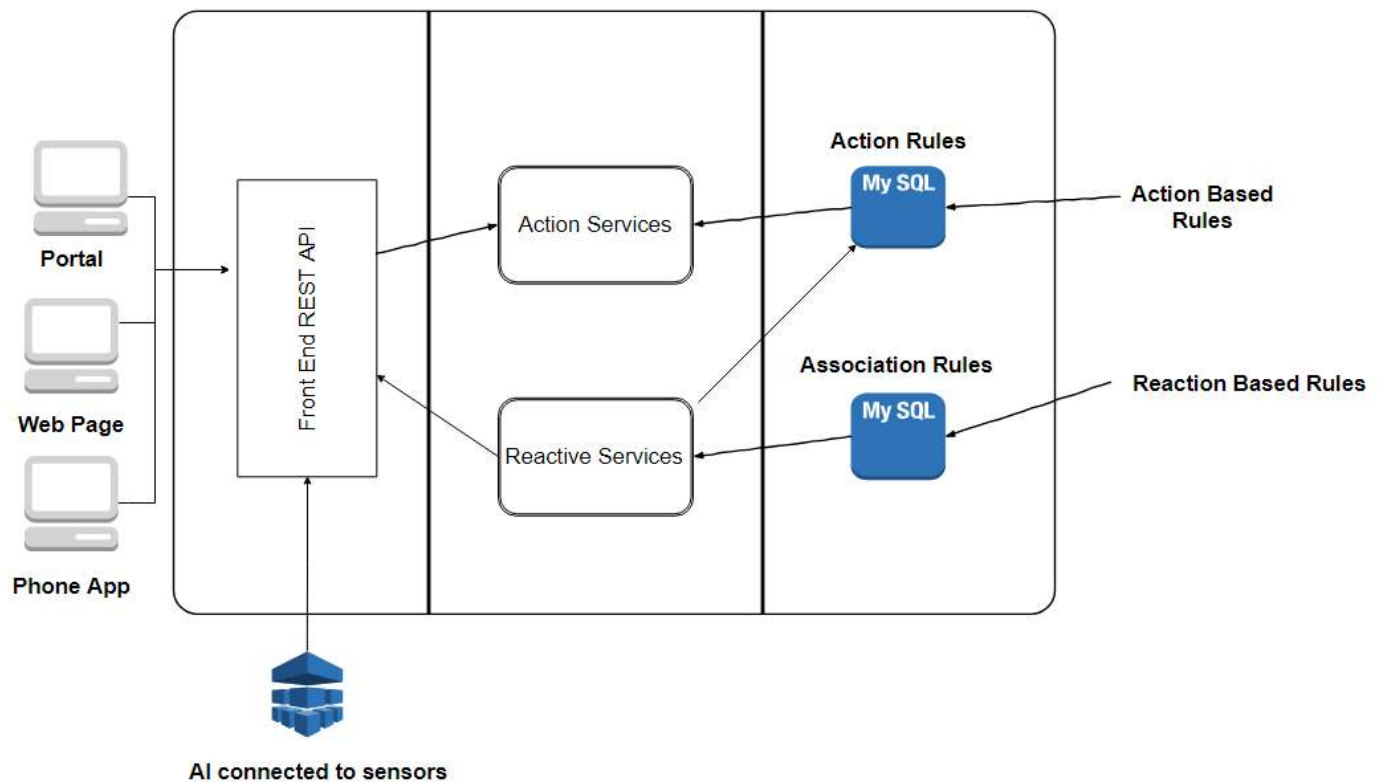
### 3. Architectural Representation

#### 3.1. Representation

##### 3.1.1. Description

3.1.1.1. PFARS will use a web-based architecture to achieve separation between the backend database services and the front-end client services. Layers will unlock services, to only those with the required credentials. Credentials are checked using role-based-access.

##### 3.1.2. Overall Diagram

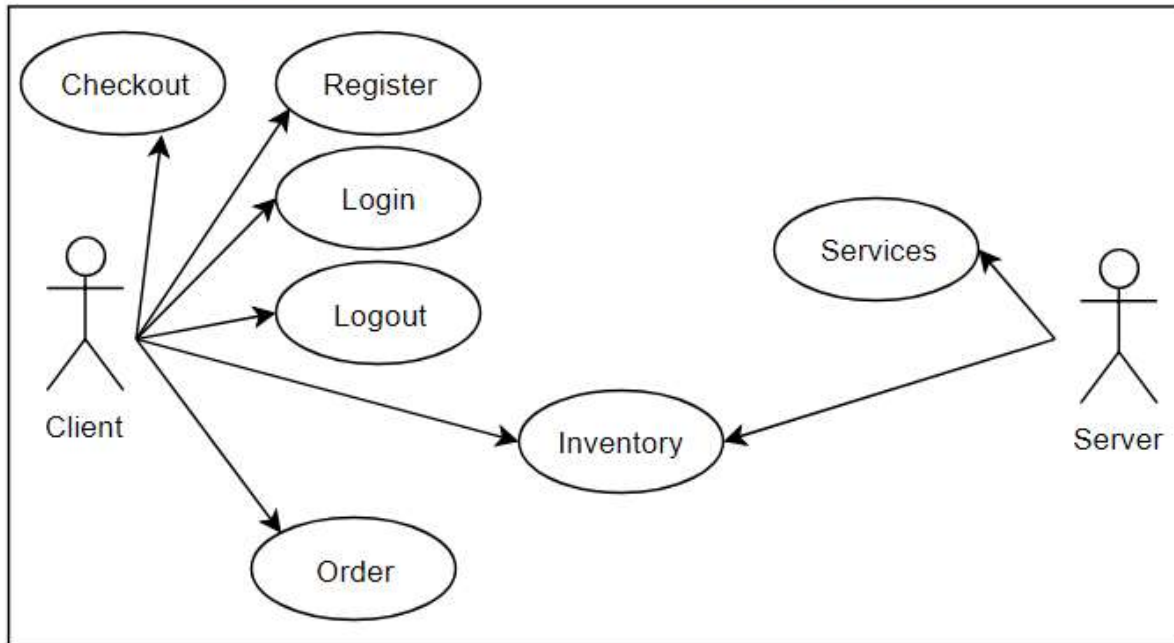


### 3.2. Views

#### 3.2.1. Use Case View

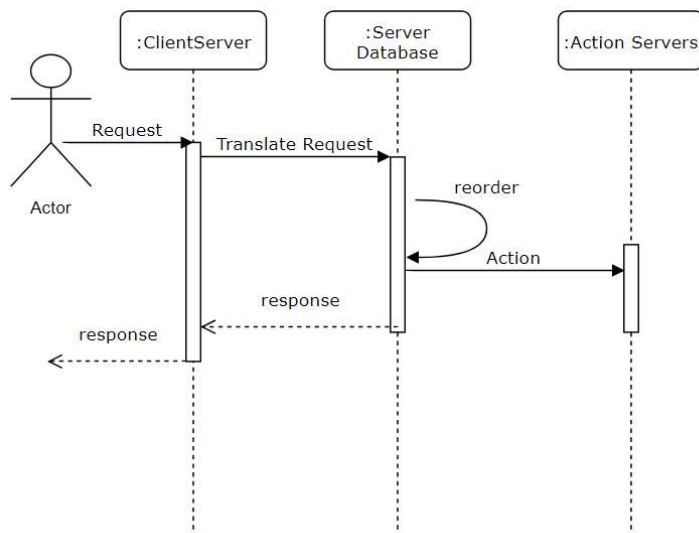
## Client Side

## Server Side



#### 3.2.2. Process View

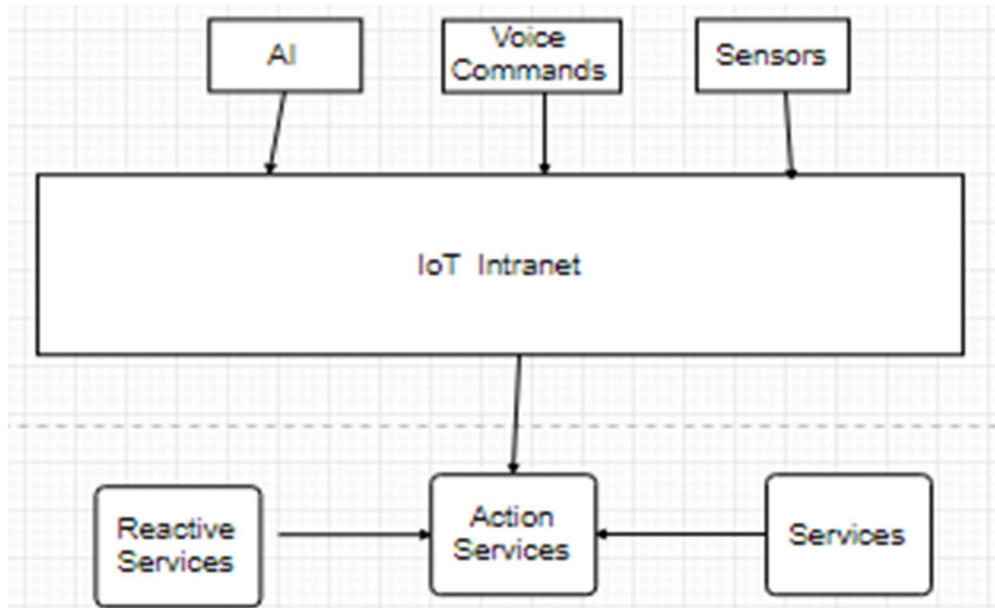
Process view shows the sequence of events in the PFARS system using Actor/Request and shows interaction between components in chronological order.





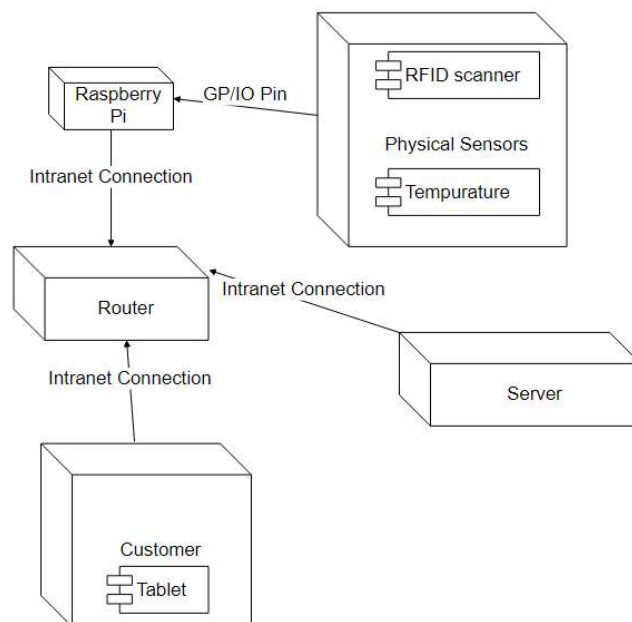
### 3.2.3. Logical View

This subsection describes the overall decomposition of the design model in terms of its package hierarchy and layers.



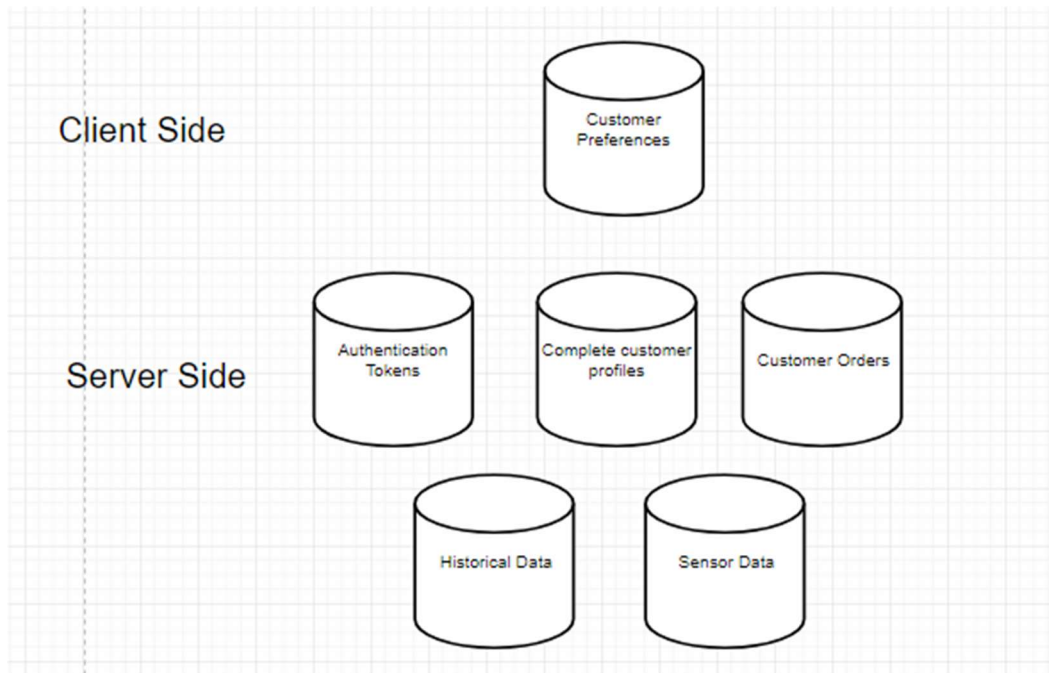
### 3.2.4. Physical View

This view shows the mapping of our main hardware components



### 3.2.5. Data View

This view outlines the storage of information of all the major data components.



## 4. Specific Requirements

### 4.1. External

#### 4.1.1. User interfaces

##### 4.1.2. Hardware interfaces

##### 4.1.2.1. Data inputs

##### 4.1.2.1.1.

##### 4.1.2.2. Data output

##### 4.1.2.2.1. Speakers

##### 4.1.2.2.2. Screens/Digital Frames

##### 4.1.2.3. Network

##### 4.1.2.3.1. Servers

##### 4.1.2.3.2. Routers

##### 4.1.2.3.3. Access points

##### 4.1.2.4. User Interface Hardware

##### 4.1.2.4.1. Smart Phones

##### 4.1.2.4.2. Hubs

- 4.1.2.4.3. Microphones and Speakers
  - 4.1.3. Software interfaces
    - 4.1.3.1. User Interfaces
      - 4.1.3.1.1. Website
      - 4.1.3.1.2. App
      - 4.1.3.1.3. Hub OS
    - 4.1.3.2. Voice Interface
      - 4.1.3.2.1. Input
        - 4.1.3.2.1.1. Microphone
      - 4.1.3.2.2. Output
        - 4.1.3.2.2.1. Speaker
        - 4.1.3.2.2.2. Screen
        - 4.1.3.2.2.3. Hub
        - 4.1.3.2.2.4. Services
- 4.2. Specific requirements
  - 4.2.1. Sequence diagrams
  - 4.2.2. Classes for classification of specific requirements
- 4.3. Performance requirements
  - 4.3.1. Network Availability 100%
  - 4.3.2. Response time, Highest priority for customers, lowest priority for inhouse maintenance and records.
  - 4.3.3. RFID response time needs to be within 2 seconds
  - 4.3.4. Voice Recognition need to be configured in 1 min
  - 4.3.5. Accessibility
  - 4.3.6. Voice recognition also acts as secondary key
  - 4.3.7. Software system attributes

- 4.3.7.1. Physically Impaired guests
- 4.3.7.2. Ramps
- 4.3.7.3. Braille
- 4.3.8. Availability
  - 4.3.8.1. System should be available 24/7
  - 4.3.8.2. System should be unavailable for time customer is not checked in or not maintenance time.
  - 4.3.8.3. Customer will also have mobile system to control utilities when outside the house
- 4.3.9. Capacity
  - 4.3.9.1. Able to service at least 20 smart homes
  - 4.3.9.2. Able to service at least 200 application
- 4.3.10. Recoverability
  - 4.3.10.1. System should be able to automatically recover vital security functions from backup in case of server failure.
- 4.3.11. Reliability
  - 4.3.11.1. System should make multiple backup copies of itself, incase system fails.
  - 4.3.11.2. In case of failure switch to different server to be serviced
- 4.3.12. Scalability
  - 4.3.12.1. Server should be able to accommodate up to 1000 devices
  - 4.3.12.2. Network should be able to accommodate traffic of all device or prioritize the traffic
  - 4.3.12.3. System should be compatible with most sensors in market
- 4.3.13. Security
  - 4.3.13.1. Deny access to all personal data of customer.
  - 4.3.13.2. Only authorize users can access data.
  - 4.3.13.3. All data must be encrypted before stored to server
  - 4.3.13.4. All physical access points must be RFID authorize

- 4.3.14. Usability
  - 4.3.14.1. Easy to connect to most personal IoT devices
- 4.4. Other requirements
  - 4.4.1. System Evaluation
    - 4.4.1.1. Correctness
      - 4.4.1.1.1. Does the system reflect the requirements of the architectural document
    - 4.4.1.2. Efficiency
      - 4.4.1.2.1. Resource usage and software functionality
    - 4.4.1.3. Flexibility
      - 4.4.1.3.1. IoT expansion in domiciles
    - 4.4.1.4. Integrity/ Security
      - 4.4.1.4.1. Prevention against attacks to the resort system
    - 4.4.1.5. Interoperability
      - 4.4.1.5.1. Software and hardware integration
    - 4.4.1.6. Maintainability
      - 4.4.1.6.1. Ease of maintenance for non-technical employees
    - 4.4.1.7. Reliability
      - 4.4.1.7.1. Ensures system backups and alternative resources to continue servers in a limited environment
    - 4.4.1.8. Reusability
      - 4.4.1.8.1. Able to adapt to new technologies and repeated use
    - 4.4.1.9. Testability
      - 4.4.1.9.1. Able to meet requirements put forth by the client that also reflect in the architectural documentations
    - 4.4.1.10. Usability
      - 4.4.1.10.1. Employees and resort clients can with minimum learning operate their respected role-based system views
    - 4.4.1.11. Availability
      - 4.4.1.11.1. System availability to client, employees, and resort clients

## References

- (n.d.). Retrieved November 27, 2018, from <https://www.serversdirect.com/servers/whitebox-open>
- 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE / Institute of Electrical And Electronics Engineers Incorporated, 1998.
- Empowering the smart home. (n.d.). Retrieved November 27, 2018, from <https://www.openhab.org/>
- Foundation, R. P. (n.d.). Teach, Learn, and Make with Raspberry Pi. Retrieved November 27, 2018, from <https://www.raspberrypi.org/>
- Stringfellow, A. (2018, June 25). What is Web Application Architecture? How It Works, Trends, Best Practices and More. Retrieved November 27, 2018, from <https://stackify.com/web-application-architecture/>