Scenarios Engine

Students : Ami Goldfarb, Roma Dyment, Gal Rotenberg, Gil Ronen

Supervisor: Ofer Raanan

Introduction

The aim of the Connected Home Scenario Engine project is to provide a scenario package that will allow the definition and execution user defined scenario.
Need to develop a server side standalone engine and a basic user interface for defining and running scenarios
Scenario engine will support the following cases:

Scenario Service Registration
The scenario service registration allows registering a new service with the available events and actions the service supports.

Scenario definition by End User
The scenario definition allows end user to create scenarios from above events and action. In addition a built in conditions for running manually and at specific time/scheduling.

Objectives

- Build cloud tenant-based application which brings to end-users interactive services such remote controls over web/mobile.
- Getting involved with IoT and programming technologies.
- ‘IoT Device’ Simulating Software - intended for project presentation

Features

- Phase 1
  - Vendor (Client) side product management
    - Basic GUI (web form)
    - Product registration
      - Add/Remove Events
      - Add/Remove Actions
  - End-user (Client) side product management
    - Basic GUI (web form)
    - Device & User registration
    - Scenario management
      - Add/Remove simple logic scenarios (concurrent or/and expressions)

Phase 2

- Advances gui interface
- Advances scneario logic - complex logic including timebased (duration , start time ,end time)

Methodology

Scrum

A series of sprints which in every sprint we declare targets, build, test, create documentation and have a delivery.

- First Sprint (1w)
  - management webpage for client (vendor & end-user)
    - Vendor (web page)
      - Log in page
      - Add device
        - Add / Remove triggers
        - Add / Remove actions
    - End user (web page)
      - Log in page
      - Add device from list
        - Device registration
      - Simple scenario planner (Nice to have)

Stories

Stories:

1. Vendor sets up a Device on server
2. Customer (Client) logs-in the server (All client-related info is shown up)
3. Customer (Client) manages his Devices on server
4. Customer (Client) manages Scenarios on server
5. Engine 'manages' defined Scenarios

Stories

Story No. 1 / Vendor sets up a Device on server – *in detail…*

- Vendor registration webpage
  - Create unique id for Vendor (according to user&password)
    - User&Password oriented Log-in
    - User&Password oriented Sign-up
  - Create IoT Device
    - Show 'Device' list been defined
    - Add new 'Device':

      Form prototype:

      +Name

      +Picture

      +Description

      +Actions list

      +Events list
- Engine updates its information to DB

Stories

Story No. 2 / Customer (Client) logs-in the server (All client-related info is shown up) – in detail…

- Customer registration webpage
  - Create unique id for Customer (according to user&password)
    - User&Password oriented Log-in
    - User&Password oriented Sign-up
  - Customer's panel
    - Show added and configured (by customer) list of 'Devices'
    - Show added and configured (by customer) list of 'Scenarios'
    - Show managing options
      - Devices
      - Scenarios

Stories

Story No. 3 / Customer (Client) manages his Devices on server – *in detail…*

- Show available 'Devices' for the customer
  - Show list of existing Vendors

    ⇓

    Show list of 'Devices' provided by chosen Vendor

    ⇓

    Select a Device

    ⇓

    Configuration (regist. and conf. S/N of device) of selected Device

Stories

Story No. 4 / Customer (Client) manages Scenarios on server – in detail…

- Customer managing scenarios for Engine
  - Add a Scenario

    Show up all customer's Devices

    ⇓

    Select a Device from shown up list

    ⇓

    Provides information of Device's supported Events. Furthermore, possibility of selection and configuration is available as well

    ⇓

    Addition of Device's Actions (continued on (*))

Stories

Continued

Story No. 4 / Customer (Client) manages Scenarios on server – in detail…

- Customer managing scenarios for Engine
  - (*) Addition of Actions to Scenario

    ⇓

    Show up all customer's Devices

    ⇓

    Select a Device

    ⇓

    Show up supported Actions of selected Device

    ⇓

    Action selection

Stories

Story No. 5 / Engine 'manages' defined Scenarios – in detail…

- Story (device fire up an event)

  Device event is triggred

  ⇓

  Device notifies the server

  ⇓

  The server finds mentioned scenario in which the device appears

  ⇓

  Updates the according flags

  ⇓

  Runs the boolean function which represents the scenario logic

  ⇓

  If the return is true => List of actions invoked

Problems, Issues and Workarounds…

- Lack of coordination between Engine Logical part and DB

- Combination between Tomcat, MYSQL and Maven lead to difficulties in using the jdbc connector.
- *In more details*: Maven should provide the connector (exclusively) and Class.from should add the explicit relation. Insufficient information on the web and from exceptions sums up into long debugging session.

Problems, Issues and Workarounds…

- Enormous dependencies needed for operation - hassle in having them all.
- *In more details:* Maven solves exactly this issue.

- Not paying close attention to specification document, lead to wrong implementation.. which changed the schedule drastically
- **Prerequisite Knowledge** : lack of prior knowledge and experience in web development.

Milestones

- **Requirements & Technology Research (Weeks 1-2)**
  - Deliverables: requirements document & technology selection
  - Start High Level Design
- **Design (Week 3)**
  - Deliverables: design document.
  - Start of coding
- **Coding Phase 1 (Week 5-6)**
  - Deliverable: mid term demo.

❑ **Mid term presentation (Week 6) – Dec 31st 2017**
  - Demo of phase 1 working features
  - 10:15 – meet mentors
  - 11:00-12:00 – each group 20 min presentation

❑ **Finalize Coding+ Testing (Weeks 7-12) – Feb 4th 2018**
  - Deliverables: code + code documentation.

❑ **Documentation (Week 13)**
  - Final presentation
  - Deliverables: Installation guide + User guide.

Development Environment

Frontend :

Languages : HTML, CSS , JavaScript (EC 6)
Tools : SASS (css pre processor), Nodejs (npm as package manager), Webpack (module bundler), Babel (js transpiler),

Libraries: bootstrap (css framwork), React.Js, Redux.js , axios (HTTP client),

Editor: sublime-text

Version Control:

GitHub

Backend:

Languages : Java, SQL

Tools: Maven, Git, Postman (chrome Ext.), SQL Workbench

IDE : Eclipse

Libraries: Jersey (REST client), Gson (Json de/serializer)

Server : Tomcat, MySQL, AWS RDS (Amazon DB), Rest.

Final Deliverables

- Software
  - Well documented source code
- Documentation
  - Release Notes
    - known bugs, caveats, workarounds.
  - Installation Guide
  - User Guide.